12-2020

Electromagnetic Interference Estimation via Conditional Neural Processing

Edgar E. Gomez

Electromagnetic Interference Estimation via Conditional Neural Processing

THESIS

Edgar E. Gomez

AFIT-ENG-MS-20-D-006

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-20-D-006

Electromagnetic Interference Estimation via Conditional Neural Processing

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Edgar E. Gomez, B.S.Cp.E.

November 27, 2020

AFIT-ENG-MS-20-D-006

Electromagnetic Interference Estimation via Conditional Neural Processing

THESIS

Edgar E. Gomez, B.S.Cp.E.

Committee Membership:

Maj Joseph A. Curro, Ph.D
Chair

Lt Col James W. Dean, Ph.D
Member

Richard K. Martin, Ph.D
Member

Abstract

The goal of this thesis is to determine the efficacy of employing Machine Learning (ML) to solve Joint Urgent Operational Need (JUON) CC-0575, which aims to develop a Common Operating Picture (COP) of the Global Positioning System (GPS) Electromagnetic Interference (EMI) environment. With the growing popularity of Artificial Neural Networks (ANNs), ML solutions are quickly gaining traction in businesses, academia and government. This in turn allows for problem solutions that were previously inconceivable using the classical programming paradigm. This thesis proposes a method to develop a COP of the battlefield via ANN ingestion of multiple-source signals and sensors.

We conduct three separate experiments with varying amounts of EMI interference sources (single, double, and triple jammer datasets). The type of ANN developed to address this problem is a Conditional Neural Process (CNP) with residual connections. The model is developed to provide the estimated EMI environment as well as a measure of confidence in its estimates, as the specific application of this model could lead to loss of life in the event the model estimates are taken as truth. The model resulted in an EMI estimator that was neutral on the single jammer test data set, yet aggressive on the multiple jammer test data sets.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Electromagnetic Interference Estimation via Conditional Neural Processing

I. Introduction

1.1 Problem Background

The goal of this thesis is to determine the efficacy of employing Machine Learning (ML) to solve Joint Urgent Operational Need (JUON) CC-0575. A JUON is a need prioritized by a combatant commander and is defined as a need requiring a solution that, if left unfilled, could result in the loss of life and/or prevent the successful completion of a near-term military mission [1]. JUON CC-0575 specifically calls for development of a Common Operating Picture (COP) for the battlefield. The COP will provide an Electromagnetic Interference (EMI) Global Positioning System (GPS) capability that will integrate multiple-source signals and sensor data with a visualization capability. This tactical capability will inform mission planners and combat operators in an air or land domain to avoid degraded tactical mission planning deficiencies resulting from unanticipated EMI events. In order to address JUON CC-0575, we will develop a machine learning solution, specifically an Artificial Neural Network (ANN), to integrate multiple-source signals and sensor data with the intent to develop a battlefield COP.

1.2 Research Objectives

At the time of this writing the Joint Navigation Warfare Center (JNWC) is developing a solution for JUON CC-0575. The objective of this thesis is to develop a framework for implementing ANNs to integrate multiple-source signals and sensor

data from a helicopter traversing an EMI challenged environment in order to develop a COP of the battlefield. The aforementioned framework embodies various modern techniques such as residual networks and Conditional Neural Processes (CNPs). In addition to developing a COP of the battlefield, visualization of the confidence of the ANN has been added as an additional objective.

1.3 Document Overview

This thesis is organized as follows:

Chapter II is a review of information and literature that is pertinent to understanding various concepts that this thesis is composed of. Chapter III is the methodology chapter of this thesis, explaining dataset generation, ANN architecture, ANN training, before ending with methods of analysis. Chapter IV applies the previously discussed ANN framework to an EMI estimation problem. ANN training and test results are presented and discussed. Chapter V provides a summary of the results and a short discussion of possible avenues for future work as a result of this research.

II. Background and Literature Review

This literature review outlines information and literature that is pertinent to understanding various concepts that this thesis is composed of. Five main sections are presented and summarized. The first section, Section 2.1, is a background on Global Positioning System (GPS). Section 2.2 is an overview of GPS receivers. Section 2.3 discusses various GPS signal interference sources, while Section 2.4 discusses technologies to detect and locate them. Section 2.5 provides an overview of Machine Learning (ML) and Artificial Neural Networks (ANNs). The final section, Section 2.7 discusses simulation software used to generate the data used for this thesis research.

2.1 Global Positioning System

Global Positioning System (GPS) is a network of earth orbiting satellites that was originally developed by the US Department of Defense (DoD) for precise time transfer as well as improved navigation and positioning for military purposes. The first satellite was launched in 1978, while the system was declared fully operational in April of 1995 [2]. The GPS is comprised of three segments: the space segment, the control segment, and the user segment. Both the space and control segments are controlled by the DoD and are responsible for the satellites themselves, as well as the management of satellite operations. The user segment covers the research and development of military and civil GPS user equipment.

2.1.1 Space Segment

The space segment is comprised of a constellation of satellites that are distributed throughout six orbital planes inclined at 55 degrees relative to the equatorial plane. Each orbital plane has four slots distributed unevenly to house primary satellites, as

well as additional slots for spare satellites as seen in Figure 1.

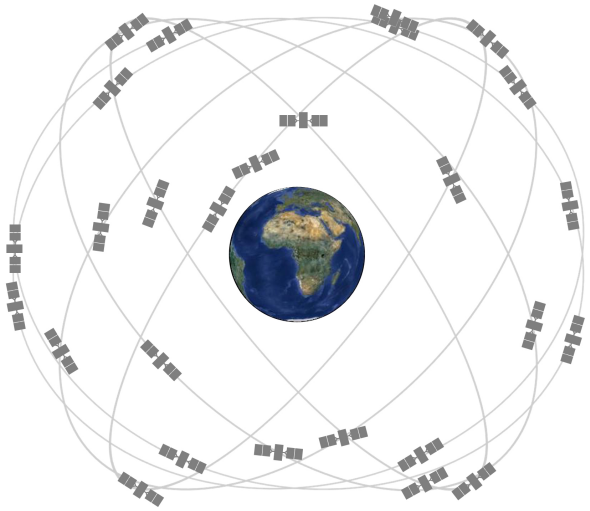


Figure 1: Expandable 24-Slot GPS Satellite Constellation [3]

The current operational GPS constellation is a mixture of legacy and modern satellites launched between the years 1997-2020 [3]. These satellites continuously broadcast extraordinarily weak earth-bound ranging signals. Table 1 summarizes the received minimum Radio Frequency (RF) signal strength for Block IIA, IIR, IIR-M, IIF, and III satellites with respect to a 20.46MHz bandwidth.

Table 1: Received Minimum RF Signal Strength for Currently Operational Satellites (20.46MHz Bandwidth) [3]

SV Blocks	Channel	Signal	
		P(Y)	C/A or L2C
IIA/IIR	L1	-161.5 dBW	-158.5 dBW
	L2	-164.5 dBW	-164.5 dBW
IIR-M/IIF	L1	-161.5 dBW	-158.5 dBW
	L2	-161.5 dBW	-160.0 dBW
III	L1	-161.5 dBW	-158.5 dBW
	L2	-161.5 dBW	-158.5 dBW

2.1.2 Control Segment

The control segment is a global network of ground-based monitoring facilities that provide the following services: Space Vehicle (SV) orbit monitoring, SV health monitoring and maintenance, GPS time maintenance, SV ephemerides and clock parameter predictions, SV navigation message updates, and SV orbit adjustments [4]. The Master Control Station (MCS) is located at the Schriever Air Force Base and provides all GPS command and control functions. The Air Force has several monitoring stations spread around the globe, allowing operators to view each SV from at least two monitoring stations continually.

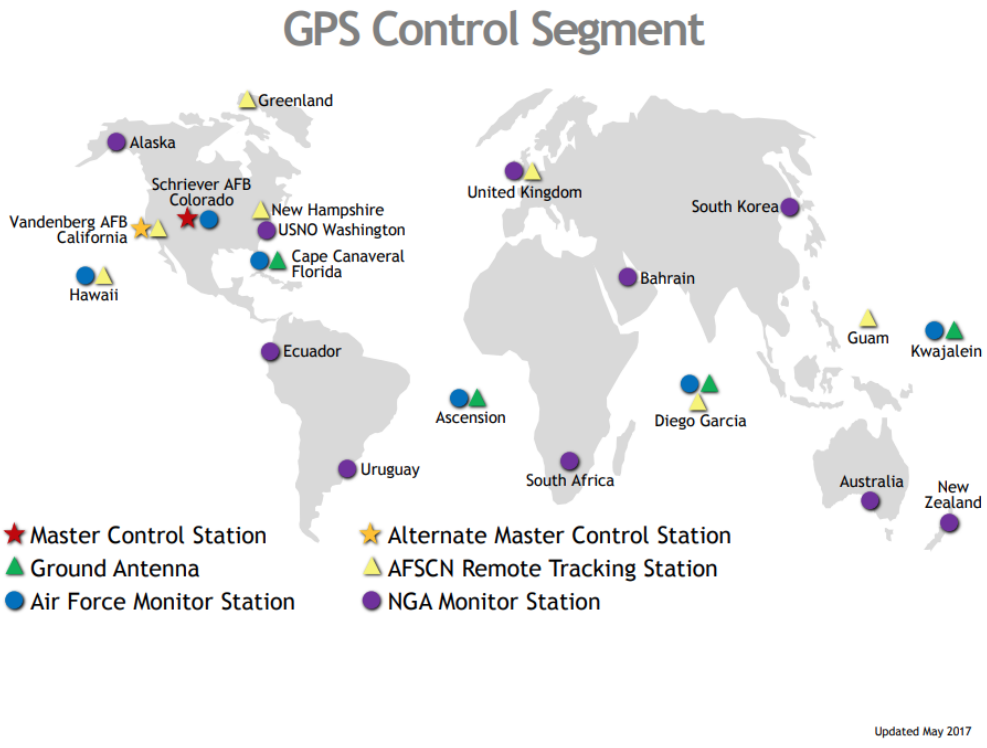


Figure 2: Control Segment Layout [4]

2.1.3 User Segment

The user segment is composed of the military and civil GPS receivers that ingest the signals broadcast from satellites to derive a Position Navigation and Timing (PNT) solution. Receivers are a very complex combination of hardware and software subsystems that are used to extract a signal from beneath the noise floor and perform signal processing to produce a position solution to the user. GPS receivers are one-way communication devices, only tracking signals transmitted by satellites, and never transmitting signals back to the satellites. The accuracy of a typical GPS receiver's solution is dependent on various capabilities and is detailed in Table 2.

Table 2: GPS receiver accuracy by capability

	Mode	Horizontal Accuracy (drms)
Stand-Alone	Civilian Receiver / With WAAS	2-3 m / 0.5-1 m
	Military Receiver (Dual Frequency)	2 m
Differential	Code Differential	1-2 m
	Carrier-Smoothed Code Differential	0.1-1 m
	Precise Carrier-Phase (kinematic)	1-2 cm
	Precise Carrier-Phase (static)	1-2 mm

The following section will go into more detail on GPS receiver design and functionality.

2.2 GPS Receiver Functionality

Global Positioning System (GPS) signals are transmitted via Medium Earth Orbit (MEO) satellites, approximately 20,200 km above the Earth's surface, and the received power is on the order of –160 dBW. This signal power can be compared to the energy received from a 25 watt light bulb from a distance of 11,000 miles away

[5]. A GPS receiver is designed to track these signals, calculate the distance to each satellite in view, and use information that is modulated onto said signal to deduce its own location. This operation is based on a mathematical principle called trilateration. Figure 3 provides an illustration of a GPS receiver functional block diagram. The remainder of this section will briefly discuss key components of the GPS receiver.

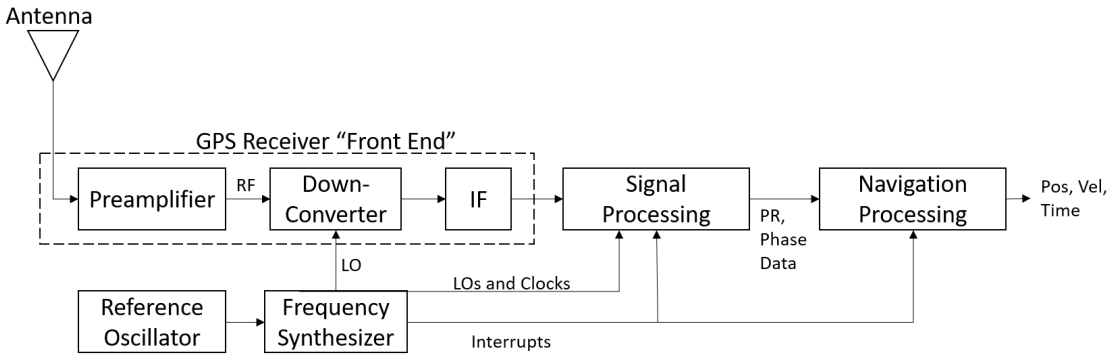


Figure 3: GPS Receiver Functional Block Diagram

2.2.1 Antenna

The antenna collects the signals transmitted by satellites and converts the incoming electromagnetic waves into electric currents. Antennas are also responsible for initial frequency filtering, that is, depending on the use case the antenna can be designed to collect only the L1 frequency, L1 and L2, or all signals, including L5. Due to the Right-Hand Circularly Polarized (RHCP) nature of GPS signals, GPS antennas are RHCP as well, which provides a degree of multi-path mitigation via rejection of Left-Hand Circularly Polarized (LHCP) multi-path signals. An important antenna parameter is directivity, which is a measure of how directional the radiation pattern of an antenna is. If an antenna has a directivity of 0 dB then it would radiate equally in all directions.

2.2.2 Preamplifier

The preamplifier is low noise amplifier, which is the first place the signal is amplified, thus setting the noise floor. It also can provide burnout protection to the rest of the receiver components downstream by means of a built-in limiter or inline filters such as pass-band and stop-band filters. The preamplifier is typically built into the antenna, but this is not required.

2.2.3 Down-converter

The down-converter is responsible for converting an input Radio Frequency (RF) signal to a predetermined Intermediate Frequency (IF) that is easier to work with from a signal processing standpoint. Down-conversion can be applied in a single stage or in multiple stages. Out of band filtering and Continuous Wave (CW) interference rejection can also be applied during the down-conversion stage.

2.2.4 Reference Oscillator and Frequency Synthesizer

The reference oscillator, often referred to as the clock, oscillates at a preset frequency and provides fundamental timing to the receiver. Many GPS receivers implement a 10.23 MHz oscillator, as that is the fundamental frequency of GPS signals. The frequency synthesizer is an electronic circuit that is implemented to generate a range of reference frequencies from the output of the reference oscillator.

2.3 GPS Signal Interference

Signal jamming, whether it be intentional or not, is one of the two largest threats when it comes to navigating using Global Positioning System (GPS) (with spoofing being the other). Signal jamming involves the use of Radio Frequency (RF) transmitters to increase the targeted environment's noise level, or to overload a receiver's

front-end electronics, ultimately resulting in the loss of signal lock. As mentioned in Section 2.1, the received RF signal strength from GPS satellites is extraordinarily weak, which means that it does not take a high-powered jammer to disrupt service.

2.3.1 Unintentional Interference

There are copious sources of unintentional interference or disruption of the GPS service provided by the United States Air Force (USAF). While not as sinister as intentional interference sources, depending on the application, the effects on the ability to utilize GPS can be detrimental. A GPS receiver can lose lock on a satellite due to an interfering signal that is only a few orders of magnitude stronger than the minimum received GPS signal strength (-160 dBW, at the Earth's surface for the L1 C/A code) [6]. In addition to the amount of power required to lose lock on a satellite, a GPS receiver requires 6 to 10 dB more carrier-to-noise density to acquire a lock than it requires for tracking [7]. The Volpe report, published in 2001, listed numerous sources of unintentional interference. Examples of these sources are as follows [5]:

- Ionospheric Interference
- Broadcast Television
- VHF Interference
- Personal Electronic Devices
- Mobile Satellite Service
- Ultra Wideband Radar and Communications

2.3.2 Intentional Interference

The growing military and civil reliance on GPS worldwide makes it an attractive target for malicious governments and groups. The US and its allies primarily navigate

using encrypted P(Y) code, but to acquire that code, most receivers must begin by tracking C/A code first. The act of disrupting or degrading GPS with the intent to deny ability to utilize the Position Navigation and Timing (PNT) service is referred to as Navigation Warfare (NAVWAR). NAVWAR is a broad term which encapsulates Electronic Support (ES), Electronic Attack (EA), and Electronic Protection (EP). One of the most popular, and most simplistic, GPS disruption techniques is the GPS Noise Jammer. The Standard Positioning Service (SPS) can be jammed over a significant area by a one watt airborne jammer. It is estimated that when airborne, this jammer can knock off an already locked GPS receiver at 10 km, and prevent it from re-acquiring at a range of 85 km [8].

2.4 Interference Detection and Geolocation

There are numerous technologies employed today that can successfully detect and pinpoint the source of GPS signal interference. These technologies often aim to safeguard critical infrastructure by performing complex monitoring of relevant signal bands to alert key personnel in real time. With one monitoring station, interference detection and directionality calculation is possible; however, to pinpoint the location of the interference source, a pre-existing infrastructure of at least three monitoring nodes is required.

One example of a GPS interference (and spoofing) detection technology is Orolia Defense and Security's BroadSense. This device comes in many form factors, ranging from a 1U rack mountable computer to a device slightly larger than a quarter. BroadSense utilizes sophisticated Global Navigation Satellite System (GNSS) receivers and patented jamming and spoofing detection algorithms to detect when the GPS signal or GPS spectrum is compromised [9].

An example of a GPS interference detection and geolocation technology is the

GNSS Interference Detection and Analysis System (GIDAS). GIDAS relies on a cluster (three or more) of monitoring stations to pinpoint the actual location of an interference source. Monitoring stations can be linked together, providing the ability to safeguard a small region of interest to an entire city or region [10].

2.5 Machine Learning

Machine Learning (ML) is a subset of Artificial Intelligence (AI) focused on producing algorithms that generate predictions based on data, while improving their accuracy over time.

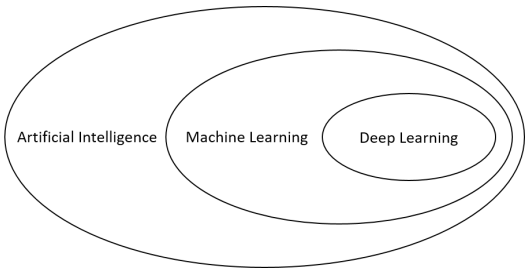


Figure 4: Artificial Intelligence, Machine Learning, and Deep Learning

ML models, unlike classical programming approaches, can produce these algorithms without explicitly being programmed to do so (see Figure 5). A ML model is trained to learn the statistical structure of many examples relevant to a task, eventually allowing the system to define the rules in which to automate the task [11]. That is, given input data points, the expected output, and a way to determine the performance of the model, an algorithm can be derived which relates the data to the appropriate output.

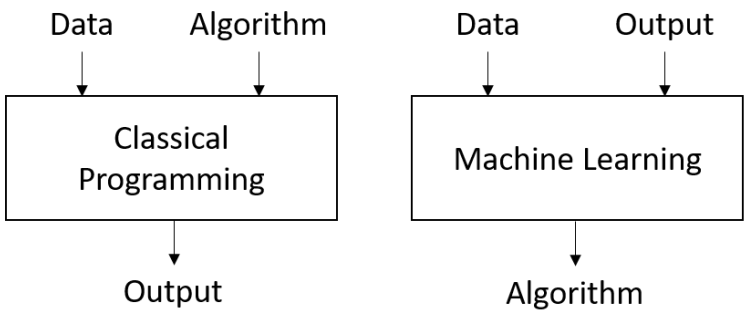


Figure 5: Machine Learning Paradigm

More specifically, a ML model assumes there is a relationship between a set of input data and all possible outputs (described in Equation (1)) and aims to learn a function f that attempts to map the input domain X onto the output domain Y.

$$Y = f(X) + \epsilon \tag{1}$$

X is the collection of input data points, Y is a set of all possible predictions, and ϵ is a random error term that is independent of X and is zero mean.

Perhaps the most straightforward example of a ML model is Simple Linear Regression. This model predicts a quantitative response Y based on a single predictor variable X, with the assumption there is approximately a linear relationship between the two [12]. The assumed linear relationship can be written as:

$$Y \approx \beta_0 + \beta_1 X \tag{2}$$

where β_0 is the bias, and β_1 is a coefficient tied to the predictor. This model aims to use training data to produce estimates for the model coefficients, denoted as $\hat{\beta}_0$ and $\hat{\beta}_1$. Once the model coefficient estimates are obtained, predictions can be made on a set of unseen input data by computing:

$$\hat{Y} \approx \hat{\beta}_0 + \hat{\beta}_1 X \tag{3}$$

where $\hat{Y}$ indicates a prediction of Y on the basis of X. While a ML algorithm can be trained to perform well, it is not safe to assume that the model created is a perfect closed form solution to our problem. Due to this, ML models are often trained to perform adequately enough on a specific problem.

2.6 Deep Learning and Artificial Neural Networks

Deep Learning, a subset of ML (see Figure 4), is a set of techniques widely employed for learning via the use of Artificial Neural Networks (ANNs). ANNs are often referred to as a 'biologically-inspired' programming paradigm enabling computers to learn via observational data. Deep learning and ANNs are particularly good at providing solutions to problems such as image and speech recognition, natural language processing, and continuous function approximation [13]. ANN are comprised of units named perceptrons and typically consist of three different types of layers: the input layer which contains an input feature vector; the output layer that reflects the ANNs response; and the layers in between, often referred to as hidden layers, that contains the perceptrons that connect the input and output. An example of an ANN is illustrated in Figure 6.

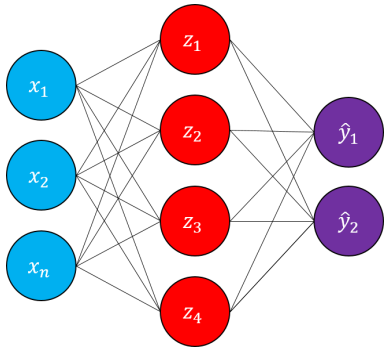


Figure 6: Single Layer Neural Network

2.6.1 Perceptron

As previously mentioned, ANNs are well suited for approximating continuous functions. This capability is achieved using ML models meaningfully constructed with perceptrons, the basic building blocks of ANNs. Perceptrons have many inputs and only one output. Figure 7 shows the basic design of a perceptron, while Equation (4) provides the mathematical representation for the perceptron.

$$\hat{y} = g\left(\sum_{i=0}^{N-1} x_i w_i\right) \tag{4}$$

where N is the number of perceptron inputs, $\hat{y}$ is the perceptron output, x_i is the i^{th} perceptron input, w_i is the i^{th} perceptron weight, and g is the activation function.

To allow for multiple inputs for X, Equation (4) can be rewritten as:

$$\hat{y} = g(w_0 + XW^T) \tag{5}$$

where $X = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$ and $W = \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix}^T$ thus allowing for the application of the activation function to the sum of w_0 and the result of a single dot product in order to solve for the output.

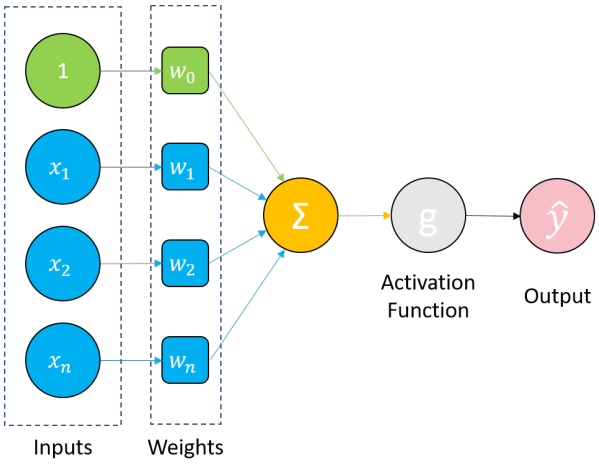


Figure 7: Perceptron Graphical Representation

Figure 8 offers a simplified representation of the perceptron, where x_i are the perceptron inputs, g is the activation function, y is the perceptron output, and z is as follows:

$$z = w_0 + \sum_{i=1}^{N} x_i w_i \tag{6}$$

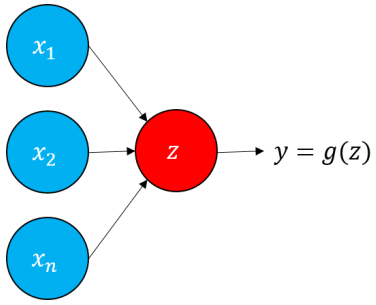


Figure 8: Simplified Perceptron Representation

2.6.2 Activation Functions

From Figure 7 it is clear to see that an activation function is used to map an input to an output. More specifically, a perceptron calculates the weighted sum of inputs, and then applies an activation function that converts it into the output. There are two main classes of activation functions: linear activation functions and non-linear activation functions. An example of a linear activation function is:

$$g(x) = x \tag{7}$$

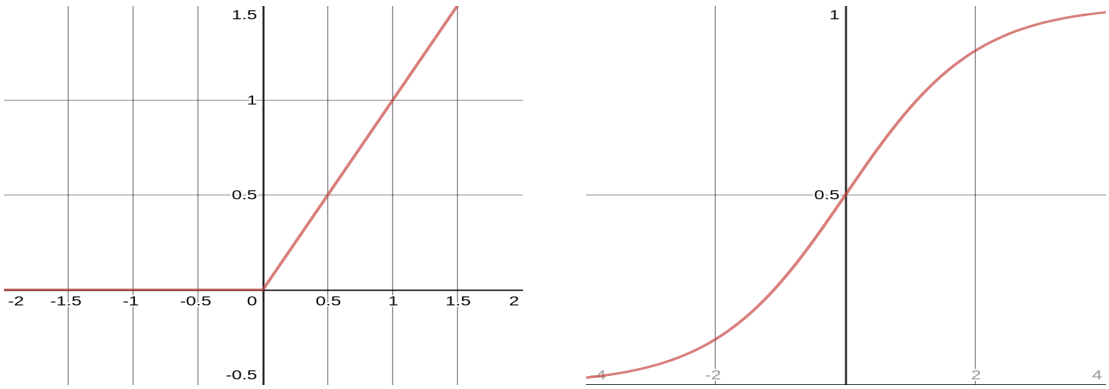
It should be noted that an ANN constructed solely with linear activation functions can only represent linear models, and is no more powerful than a multivariate linear regression. The true power of an ANN is achieved when using non-linear activation functions, as this will allow for the introduction of non-linearities into the neural network. The non-linearities introduced by non-linear activation functions are

paramount in a neural network's ability to learn complex relationships and patterns in data. Two examples of non-linear activation functions are the Rectified Linear Unit (ReLu) and Sigmoid functions. A ReLu (see figure 9a) is a function meant to zero out negative values, and is mathematically described as:

$$ReLu = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{8}$$

whereas Sigmoid squeezes arbitrary values into the [0,1] interval (see figure 9b), thus outputting a value that is able to be interpreted as a probability, and is mathematically described as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{9}$$



(a) ReLu Activation function (b) Sigmoid Activation Function

Figure 9: Examples of Non-Linear Activation Functions

2.6.3 Loss Functions

Loss functions, denoted as $\mathcal{L}$, are used to measure the cost associated with incorrect predictions that an ANN may produce. The main objective of training an ANN is to minimize the loss, meaning the smaller the loss, the better. The loss represents

a measure of success in both regression and classification problems. Regression problems aim to approximate a mapping function from an input variable (or set of input variables) to a continuous output variable (or set of output variables). Due to the nature of regression problems, a popularly employed loss function is Mean Squared Error (MSE):

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2 \tag{10}$$

where $\hat{f}(x_i)$ is the prediction that $\hat{f}$ produces for the i^{th} observation [12]. The MSE will be small if the model's response is close to the true response, and will grow as the model's response and true response begin to differ significantly.

2.6.4 Residual Connections

In traditional ANNs each layer feeds into the subsequent layer, until reaching the output layer. In residual ANNs, each layer feeds into the next layers and select layers may also feed into predetermined downstream layers. Before providing additional detail, Figure 10 provides an illustration of this concept.

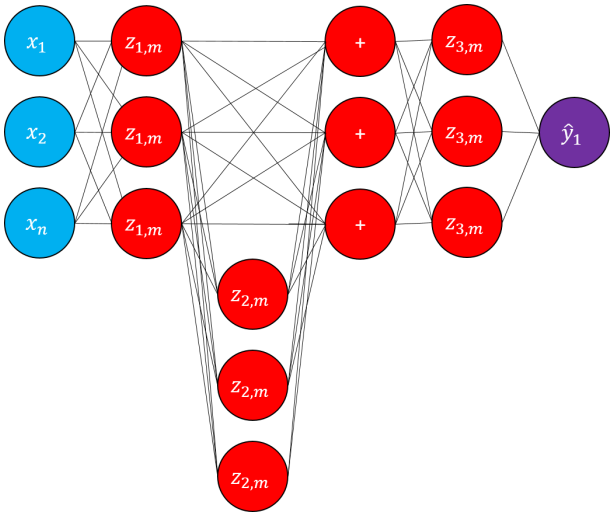


Figure 10: Residual Network Representation

Residual connections were built on the assumption that since multiple nonlinear

layers can asymptotically approximate complicated functions, then it must also be true that they can asymptotically approximate the residual functions [14]. These residual connections are used to allow gradients to flow directly through an ANN, bypassing whole layers that may cause the gradients to explode or vanish. Residual connections also allow gradients to flow throughout an ANN in reverse, which is useful when performing back-propagation in order to aid in minimizing the loss function.

2.6.5 Conditional Neural Processing

While deep neural networks are great for approximating continuous functions, they must be retrained for each new function of interest and typically require large training datasets. Gaussian Processes (GPs) can be trained to infer the shape of a new function, but are computationally expensive and quickly become intractable as the dataset and/or dimensionality increases [15].

Conditional Neural Processes (CNPs) aim to combine the flexibility of GPs while implementing them as ANNs that are trained via gradient descent. More specifically, CNPs are models that directly parametrize conditional stochastic processes. This is achieved by parametrizing the mean and variance of a Gaussian distribution for every target data point. One of the most beneficial properties of CNPs is that they are flexible with respect to target input values. That is, the model is able to be queried at resolutions it has not been trained for. This property allows for enhanced and simplified scalability of CNP implementation.

2.7 Systems Tool Kit

Systems Tool Kit (STK) is a commercial modeling and analysis software application developed and maintained by Analytical Graphics, Inc (AGI). STK is used for modeling, analyzing, and visualizing complex systems along with their sensors and

communication links, in context of the mission environment.

2.7.1 Scenarios

Each analysis space created in STK is called a scenario. A scenario is a collection of any number of objects selected by the user that will interact with each other and the mission environment during the simulation. Each scenario has well-defined temporal limits for each child-object as well as base units and various other properties. While only one scenario may exist at any time, data can be exported and reused as necessary [16].

2.7.2 Objects

At the heart of the STK analysis engine are Objects. Objects are inserted into scenarios, allowing for the generation of reports and graphics detailing the relationship between two or more Objects. A few examples of Objects are satellites, aircraft, ground vehicles, ships, planets, stars, and communication links.

2.7.3 Reports

The STK analysis engine models and analyzes the complex physical layer relationships among all Objects (and the mission environment) in a Scenario. STK provides this data to the user via Reports which are text files that detail data such as received Radio Frequency (RF) power through isotropic antenna, vehicle orientation and position, and environment power.

2.7.4 External Control and Automation

STK can be launched and commanded by any Windows program that can serve as an automation client. An automation client can be developed in various languages

such as Visual Basic, C++, VBScript, Perl, Python, and more [17].

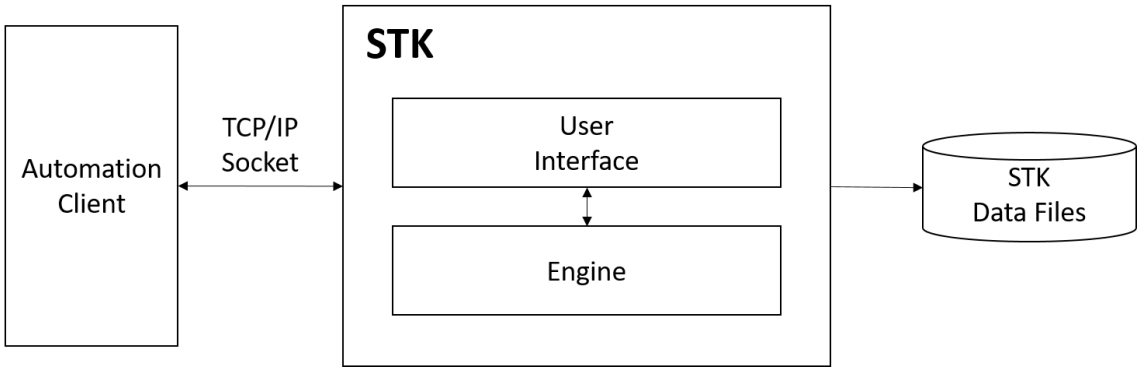


Figure 11: STK External Control and Automation Diagram

The purpose of the automation client is to obtain the current status of STK, send commands to the STK server, and interpret responses before finally issuing the next appropriate command. When STK is launched a socket connection is opened on port 5001. STK will then listen for commands issued over a TCP/IP socket that the automation client bind is meant to bind to. External control allows for the automation of repetitive tasks which helps to eliminate the possibility of command syntax errors from the user while tens of thousands of scenarios are run for days, weeks, or months on end.

III. Methodology

The goal of this research is to develop and train a single model that implements a Conditional Neural Process (CNP) to predict the electromagnetic environment, given a small subset of context points. The model will output a tensor containing the parameters of a normal distribution (a mean and a standard deviation) for each queried target point. Thus, the model will output the mean of the predicted signal-to-noise ratio as well as the standard deviation for that distribution. The smaller the standard deviation, the more confident the model is in the answer. Once a model has been developed, performance will be analyzed by observing the mean and variance of the predicted response, as well as the Mahalanobis Distance (MD).

The remainder of this chapter will elaborate on details regarding the data simulation and collection, Artificial Neural Network (ANN) architecture, ANN training, and model evaluation / analysis strategies.

3.1 Data Simulation and Collection

It is important to note that all data used in this research was simulated using Systems Tool Kit (STK). STK is used to perform high fidelity modeling of radio interference signal propagation and path loss which will include a latitude, longitude, and altitude, as well as the total Radio Frequency (RF) power in Decibel Watts (dBW). STK will also compute access reports among the receive antenna of the aircraft and the jammers scattered throughout the mission environment, which will include access time and total RF power in dBW. Finally, STK will compute the positional data of the aircraft at each time step such as Lat, Lon, Alt, Yaw, Pitch, and Roll.

Three different scenarios were simulated: A single interference source, double

interference sources, and triple interference sources. This section will describe the nature of each simulated scenario, the type of data that was collected from the simulations, and the collection automation procedures.

3.1.1 Scenario Description

There are three scenarios that are modeled in STK for this research: The single jammer scenario; the double jammer scenario; and the triple jammer scenario. Each scenario has a predefined starting Area of Interest (AOI), date-time, environment, data sampling rate, and vehicle model (STK object). While each scenario inherits key properties from the base mission environment, there are also various elements that are unique between each scenario and each scenario run as well such as aircraft dynamics, jammer location and orientation, and duration.

Although there are three different datasets simulated, each generated dataset included one helicopter and anywhere between one to three jammers. The helicopter provides Latitude, Longitude, Roll, Pitch, and Yaw measurements via an Embedded Global Positioning System (GPS) and Inertial Navigation System (INS) (EGI). There is also a spectrometer that provides in-band power measurements within a 25 MHz bandwidth centered at GPS L1 frequency, expressed in dBW. The jammers are equipped with 13 Decibel Isotropic (dBi) directional helix antennas through which a 1 MHz L1 jamming tone is transmitted.

3.1.1.1 Assumptions

This section lists the assumptions made while performing modeling and data simulation:

- No Body Shading - We assume there is zero signal propagation error due to the installed antenna pattern platform body shading effects.

- The transmitted interference signal is steady and does not display any variations in spatial or temporal dynamics (three-dimensional rather than four-dimensional interference).

- The helicopter receive antenna is isotropic, with no body or collection antenna effects.

- There are no terrain effects causing propagation error of additional power loss between the jammer transmit antenna and the helicopter receive antenna.

3.1.1.2 Interference Sources

For each scenario the helicopter will traverse a randomly generated route while producing an access report on the communication link between helicopter's receive antenna and each jammer facility in play. This access report will provide time ordered samples of the received signal power level at the front end of the installed antenna. Contour maps showing representative examples of the jammer lay-downs can be seen in Figure 12, Figure 13, and Figure 14.

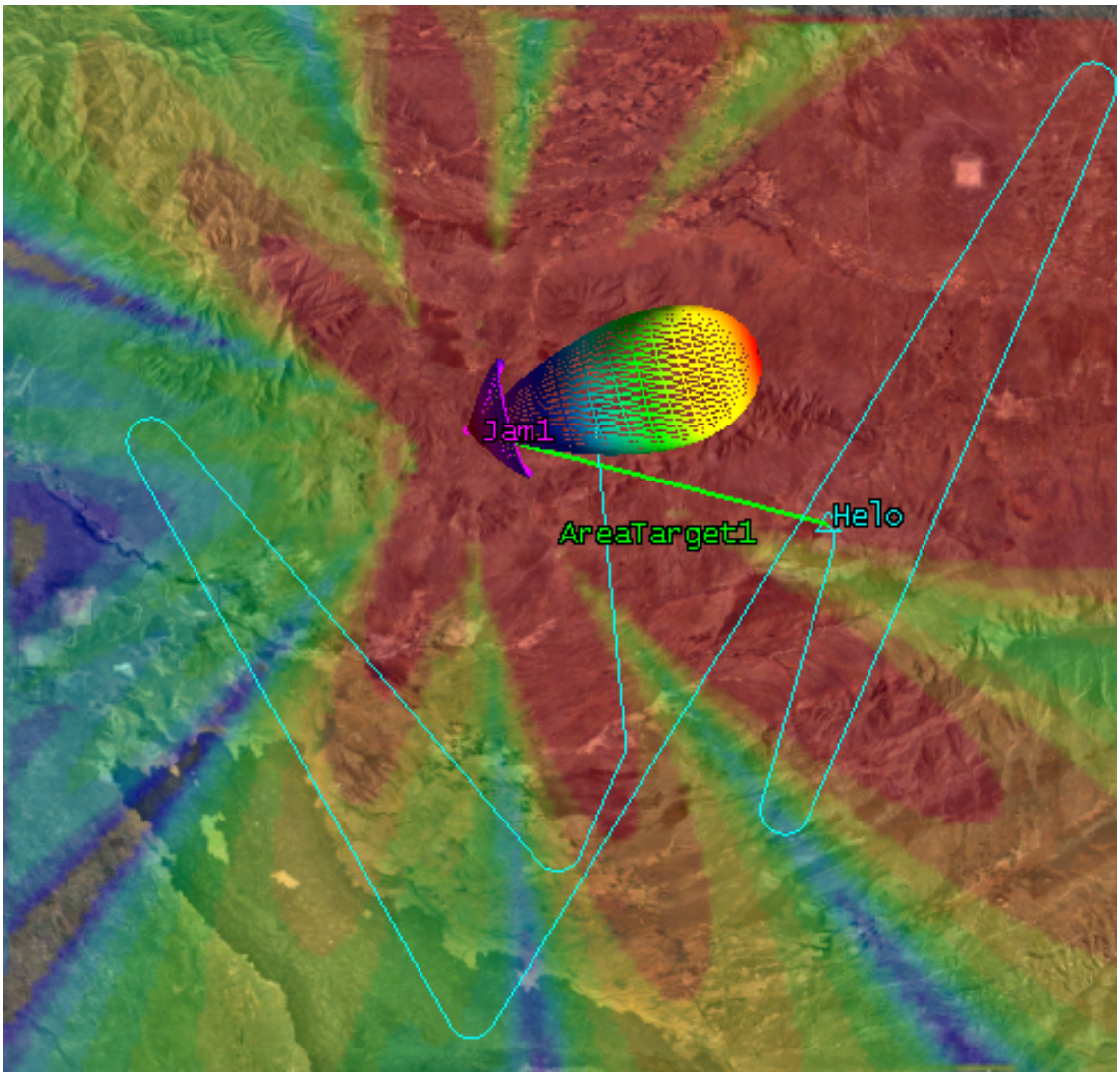


Figure 12: Contour plot showing the Electromagnetic Interference (EMI) environment with respect to a single interference source. The color map overlay ranges from blue where there is minimal EMI, to red where there is substantial EMI.

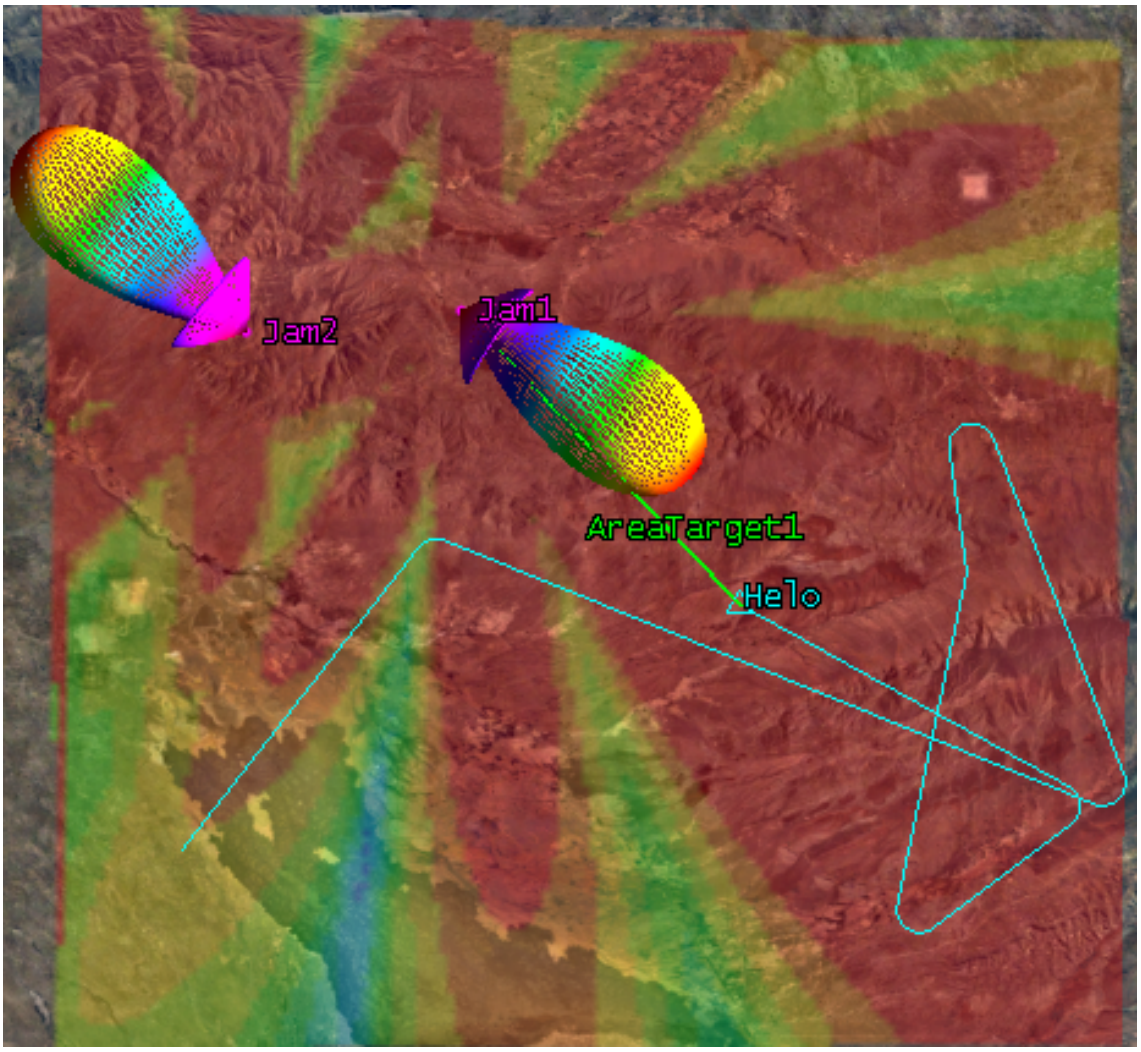


Figure 13: Contour plot showing the EMI environment with respect to double interference sources. The color map overlay ranges from blue where there is minimal EMI, to red where there is substantial EMI.

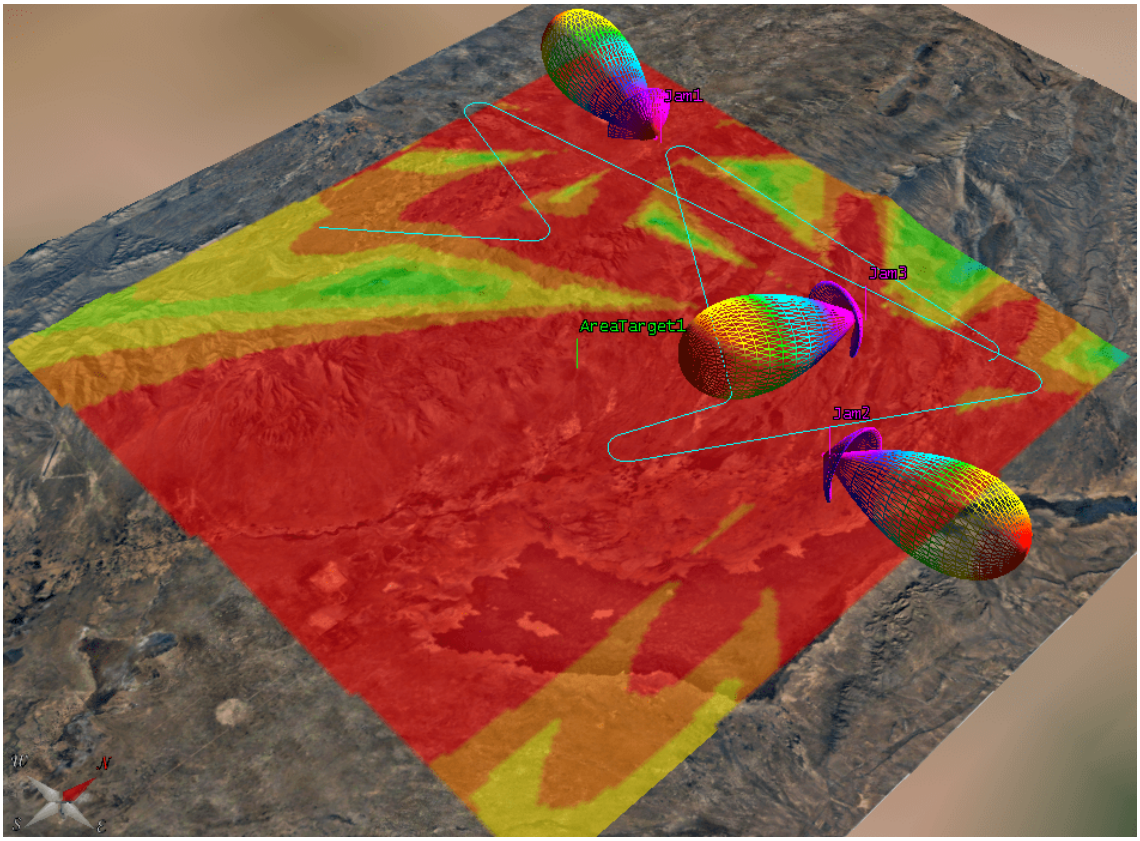


Figure 14: Contour plot showing the EMI environment with respect to triple interference sources. The color map overlay ranges from blue where there is minimal EMI, to red where there is substantial EMI.

3.1.2 WSMR Dataset

The White Sands Missile Range (WSMR) Dataset is a collection of data that details the true EMI environment, as well as the EMI environment perceived by the helicopter as it traverses a predetermined, randomly generated path (shown in Figure 15).

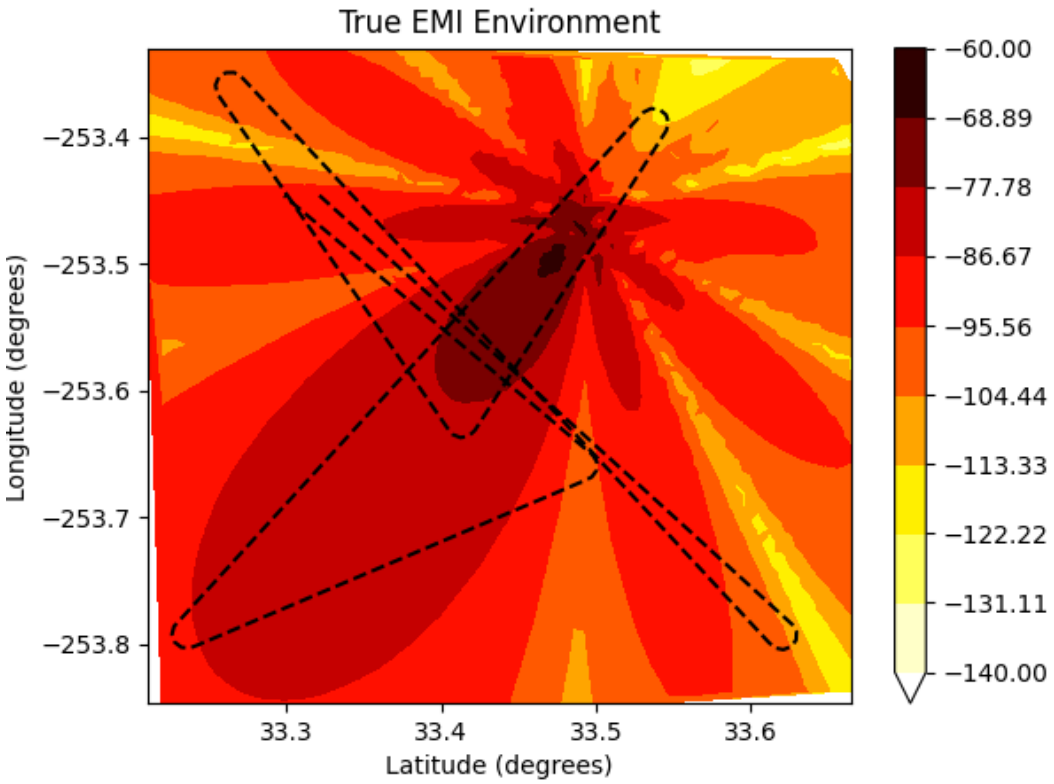


Figure 15: True EMI environment with helicopter flight trajectory overlay.

The helicopter's perceived EMI environment will be used as input features, while the true EMI environment will be output targets. Table 3 details the number of scenarios successfully simulated, which provided one set of training data each.

Table 3: Number of successfully simulated datasets by scenario

Dataset	**Number of Scenarios**
One Jammer	53,200
Two Jammers	27,442
Three Jammers	15,000
Total	95,642

Each simulated scenario results in the generation of three files; a helicopter dynamics file, the communication link description file, and the true AOI coverage file. The helicopter dynamics file (HELO_LLA_idx_HH_MM_SS.txt) describes the time trajectory and orientation of the helicopter sampled at a 0.1 Hz frequency. The communication link description file (RxISOpwr_LLA_idx_HH_MM_SS.rpt) describes the helicopter's receive antennas perceived signal power sampled at a 0.1 Hz frequency. The AOI coverage file (coverage_idx_HH_MM_SS.txt) provides the Figure of Merit (FOM) values at each grid location in the AOI. The AOI is broken up into 2832 discrete locations with a resolution of 0.01 deg between grid points. Table 4 describes the simulated datasets.

Table 4: Dataset features simulated by STK, along with the associated data type and source.

Feature Name	**Data Type**	**Source**
Latitude	$\mathbb{R}(\phi)$	GPS/INS
Longitude	$\mathbb{R}(\phi)$	GPS/INS
Yaw	$\mathbb{R}(\phi)$	INS
Pitch	$\mathbb{R}(\phi)$	INS
Roll	$\mathbb{R}(\phi)$	INS
RF Power	$\mathbb{R}$ (dBW)	Spectrum Analyzer

3.1.3 Data Collection Automation

Scenario simulation and data collection was automated using Python and the STK integration module. Without automation, a single user would have to generate tens of thousands of scenarios by hand, introducing the risk of mistakes and tainting the integrity of the dataset. By automating scenario simulation and data collection, the possibility of user error leaking into the simulation results was eliminated.

3.2 ANN Architecture

This section elaborates on the architecture of the implemented ANN model. In Section 3.1 we outlined the features generated by STK in Table 4. These features are the input features for the ANN, while the output features are the estimated mean and standard deviation of the desired targets.

The ANN model developed throughout this thesis is a CNP and residual connections. Figure 16 shows a graphical representation of the implemented model, with the inputs flowing from the top of the block diagram and producing outputs at the bottom. The ANN parameters (number of layers, number of residual connections, etc.) were chosen to reduce training loss, while the ANN hyper-parameters (layer size, learning rate, etc.) were chosen by performing a Ray-Tune hyper-parameter sweep.

The ANN begins with an input layer of size (*None*, *None*, 6). (The *None* keyword alludes to a partially known input shape, and shows that the model does not know the batch size, or how many measurements are ingested until run-time.) The input layer is where the helicopter position, orientation, and spectrum analyzer measurements are ingested by the model. The input layer then feeds into a dense layer with two outputs. The first output flows through three cascaded dense layers which then flow into an Add layer along with a residual connection from the first dense layer.

The output of the Add layer then becomes the input into another dense layer which flows into a lambda layer along with an input tensor of the targets to be queried. The purpose of the lambda layer is to tile the average output from the previous dense layer with the input tensor of targets. The lambda layer output is then concatenated with the input tensor of targets before flowing into another dense layer with two outputs. The first output flows through three cascaded dense layers which then flow into an Add layer along with a residual connection from the previous dense layer. The output from the previous dense layer has a linear activation function, with a size of (*None*, *None*, 2) which flows into the final activation layer containing a custom activation function.

The model is developed to predict the distributions (mean and standard deviation) of each latitude/longitude pair in the target input tensor. To make these predictions, the model uses the helicopters dynamics and perceived environment. However, it should be noted that the model has the ability to predict any arbitrary number of target points, and it is not confined to the target points that are used during the training sequences.

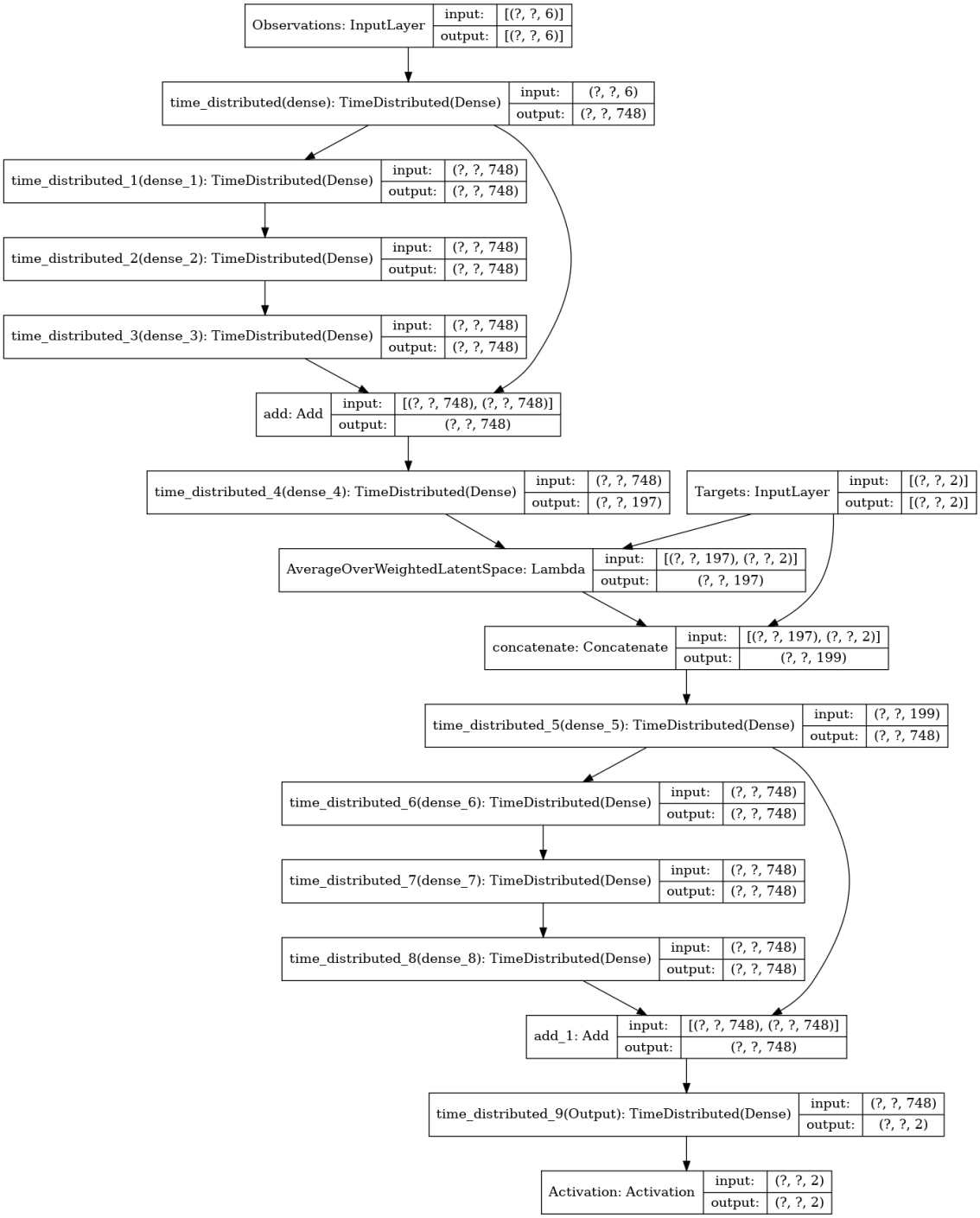


Figure 16: Graphical Representation of the Model

3.3 ANN Training

3.3.1 Custom Loss and Activation Functions

The implemented custom loss is the Negative Log Likelihood (NLL) (seen in Equation (11)) of the mean and standard deviation, while the custom metric is a mean absolute error that only uses the predicted mean and true EMI but not the standard deviation.

$$NLL(x) = -log(x) \tag{11}$$

The implemented custom activation function aims to force the standard deviation to be a positive number. To prevent the standard deviation from being zero, ELU_{plus1} is used. Exponential Linear Unit (ELU) does not suffer from the problem of vanishing or exploding gradients [18]. The piece-wise equation for ELU_{plus1} is as follows:

$$ELU_{plus1}(x) = \begin{cases} e^x & \text{for} \quad x \leq 0 \\ x + 1 & \text{for} \quad x > 0 \end{cases} \tag{12}$$

Much like ReLu, ELU slowly approaches negative one when negative, so when one is added to it will slowly approach zero when negative and not explode when larger than one. This modified ELU is what is implemented as our custom activation function.

3.3.2 Hyper-Parameter Selection

Before training the model, Ray Tune was implemented to assist in finding the optimal hyper-parameter combination. Ray Tune is a Python library designed for scalable hyper-parameter tuning [19]. The Air Force Institute of Technology (AFIT) Cyber Development Network (CDN) was leveraged to perform a 100 experiment hyper-parameter sweep to help converge on the ideal hyper-parameter combination resulting

in the minimization of the training set loss. Due to the low probability of over-fitting mentioned in Section 3.3, there was no validation split, and all data was used for training. The final set of training parameters that were chosen by experimentation are itemized in Table 5, as well as the bounds that were experimented.

Table 5: Parameter and value selection bounds for hyper-parameter sweeping and model selection.

Parameter	Lower Bound	Upper Bound
Targets	32	512
Batch Size	32	256
Initial Learning Rate	1e-6	0.001
Latent Size	16	512
Dense Units	8	1024

A Population Based Training (PBT) approach was taken while hyper-parameter sweeping was conducted. PBT allows for exploitation of previously discovered desirable hyper-parameters, dedicates more training time to promising models, and can adapt the hyper-parameter values throughout training [19]. The only inter-experiment hyper-parameter that was perturbed was the learning rate which is scheduled for reduction by a factor of 0.8 per epoch.

3.3.3 Training Details

Each scenario stripped the first 100 times steps sampled at 0.1 Hz. The reason for choosing 100 time steps is due to the variable length of the randomly generated helicopter trajectory, not all datasets have more than 100 samples available. Model training parameters were initialized as shown in Table 5 and 300 training epochs were performed with 300 steps each. Batched scenarios are randomly selected from the collection of all possible scenarios. Each epoch resulted in the model seeing 75,600

scenarios per epoch. Due to this approach there is a small probability that the model would be exposed to the same scenario multiple times in the same epoch, but the chance of being exposed to the same targets is highly unlikely and does not warrant the use of a validation set or pose a large risk of over-fitting. Model weights are updated after each batch, and a model checkpoint event is executed upon detection of an improved training loss.

We initialize the learning rate as 0.0004 and schedule reduction by a factor of 0.8 when learning stagnates. That is, if there is no improvement in the training loss after 10 epochs, the learning rate is reduced. The training sequence will then go through a cool-down phase for 10 epochs before reducing the learning rate if it is still stagnant. No early stopping is implemented, thus the training sequence will end after 500 epochs and the weights of the best model are saved to file.

3.4 Methods of Analysis

3.4.1 Mean, Variance, and Mahalanobis Distance

The model is developed to output a Gaussian mean and variance for each target, and is trained with respect to the NLL (Equation (11)) of the true scenario EMI environment. The output mean serves as the model's prediction of what the signal to noise ratio is at each target point. The output variance serves as the uncertainty in the model's prediction. The MD is a measure of the distance between a point P and a distribution D and is a generalization of the idea of measuring how many standard deviations away P is from the mean of D [20]. Given a set of observations $\vec{x} = (x_1, x_2, x_3, ..., x_N)^T$ from a set of observations with mean $\vec{\mu} = (\mu_1, \mu_2, \mu_3, ..., \mu_N)^T$ and a covariance matrix S the MD can be defined as [21]:

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \tag{13}$$

The concept of MD can be visualized in Figure 17, where it can be observed that even though observations in Y are equidistant from the mean at (0,0), the value of the MD is not the same.

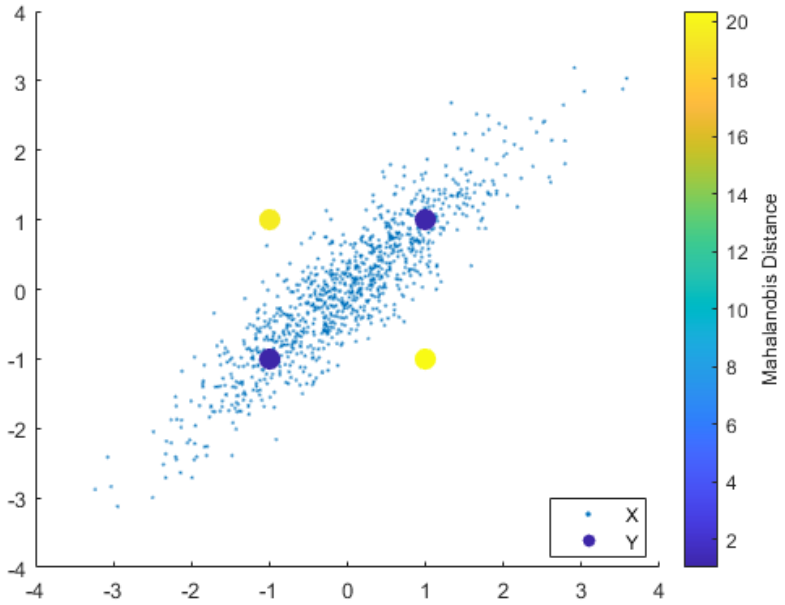


Figure 17: Each observation in Y is the same distance from the mean at (0,0). However, the MD varies (note the color scale). This is because MD considers the covariance of the data [22].

Given a covariance matrix S, which happens to be one dimensional, the squared MD reduces to the following:

$$D_M^2(\vec{x}, \vec{y}) = \frac{(x-y)^2}{s^2} \tag{14}$$

where x is an observation from a set of observations with mean y and s is the standard deviation of x and y over the sample set. It is also important to note that the squared MD follows the chi-squared distribution with n degrees of freedom, where n is the number of dimensions of the normal distribution. The probability density function for the chi-squared distribution is:

$$f(x,k) = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} x^{\frac{k}{2}-1} e^{-\frac{x}{2}} \text{ , for } x \text{ and } k > 0 \tag{15}$$

By constructing contour plots of the mean, variance, and MD the performance of this model can be analyzed. The mean contour plot (Figure 18) shows what the predicted EMI environment looks like, based off of the input features. The covariance contour plot (Figure 19) shows the model's confidence in its predictions. The MD plot (Figure 20) shows the actual performance of the model adjusted for confidence, however it must be noted that this plot is not something that can be constructed in real time, but rather is used to analyze the performance of the model on test data.

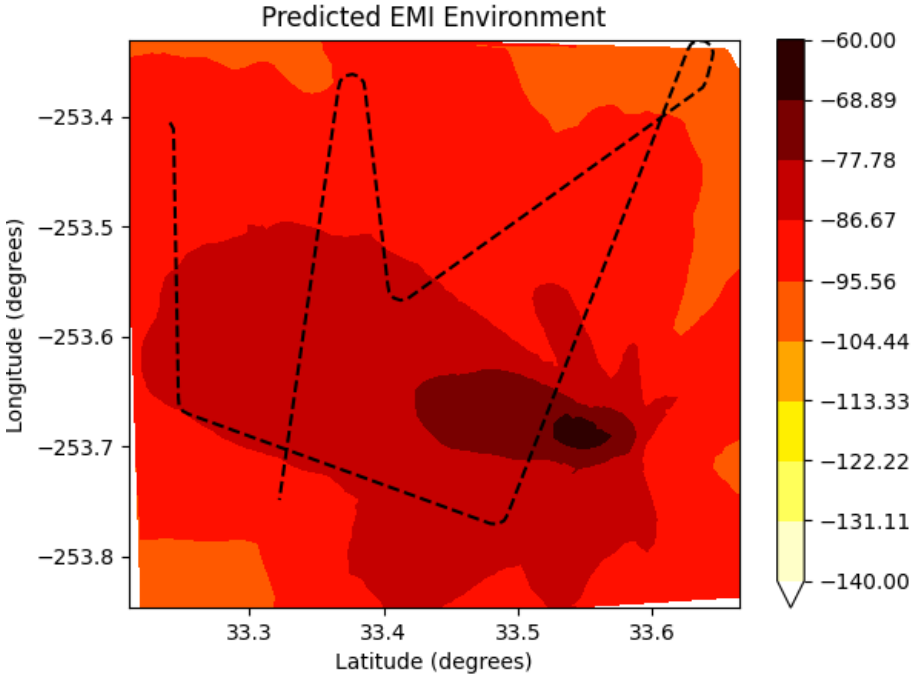


Figure 18: Example plot of the ANN's response mean.

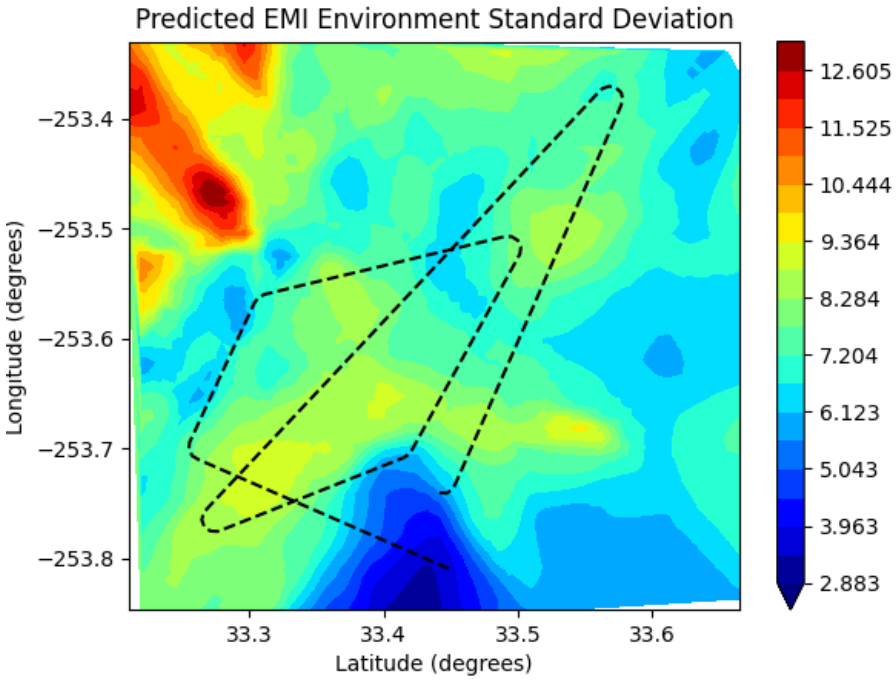


Figure 19: Example plot of the standard deviation of the ANN's response.

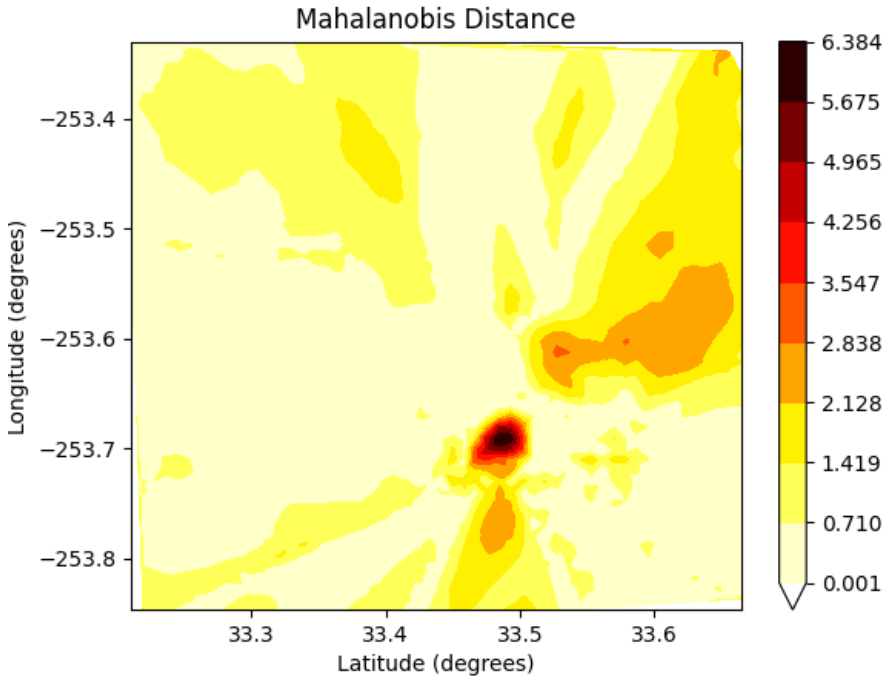


Figure 20: Example plot of the Mahalanobis Distance of the ANN's response.

3.4.2 Q-Q Plot

In the field of statistics, a Quantile-Quantile (Q-Q) plot is a probability plot, allowing for the comparison of two probability distributions by plotting their quantiles against each other [23]. The purpose of Q-Q plots is to find out if two sets of data come from the same distribution. That is, if data is believed to be Gaussian, a Q-Q plot shows just how Gaussian that data truly is. By plotting the Quantile's of two separate datasets against each other, if the two data sets come from a common distribution, the points will fall along a 45 degree reference line. An example Q-Q plot where sample observations are compared to the Standard Normal Distribution can be seen in Figure 21. If the sample quantiles do not belong to the same distribution as the theoretical quantiles the constructed Q-Q plot will portray either a bias, increased variance, or both.

To analyze the performance of the ANN, we will be comparing the squared MD (D_M^2) quantiles to those of the chi-squared distribution (χ^2). If the model is truly outputting a Gaussian distribution for each target point, then the quantiles should match, and we say the model behaves neutrally. If the D_M^2 quantiles are lower than those of the χ^2, we say the model is aggressive. If the D_M^2 quantiles are higher than those of the χ^2, we say the model is conservative.

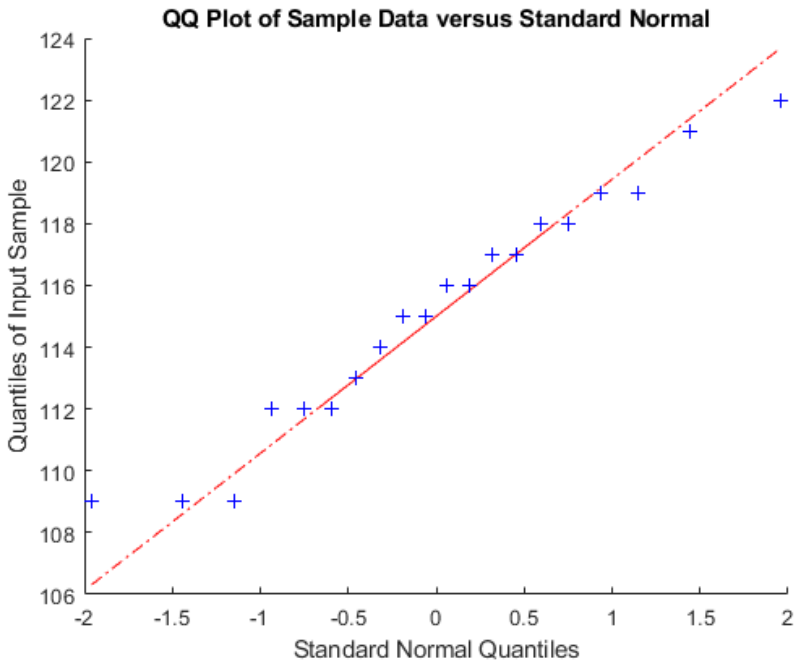


Figure 21: The Q-Q plot produces an approximately straight line, suggesting that the two sets of sample data have the same distribution [22].

IV. Results and Analysis

This chapter discusses the results of the Artificial Neural Network (ANN) presented in Chapter III. Section 4.1 and Section 4.2 will elaborate on hyper-parameter selection and training results, while section 4.3 provides a description of the test scenarios. The chapter then closes with presentation and discussion of test results for the single, double, and triple jammer scenarios in Section 4.4.

4.1 Hyper-Parameter Selection Results

After the hyper-parameter sweeping discussed in Section 3.3.2 concluded, the optimal hyper-parameters were discovered. The hyper-parameters are displayed in Table 6. Figure 22 provides a visualization of the Population Based Training (PBT) results on a per-experiment basis.

Table 6: Parameter and value selection bounds for hyper-parameter sweeping and model selection. The rightmost column reflects the optimal set of parameters chosen by experimentation across 100 trials.

Parameter	Lower Bound	Upper Bound	Optimal Value
Targets	32	512	504
Batch Size	32	256	252
Initial Learning Rate	1e-6	0.001	0.0004
Latent Size	16	512	197
Dense Units	8	1024	748

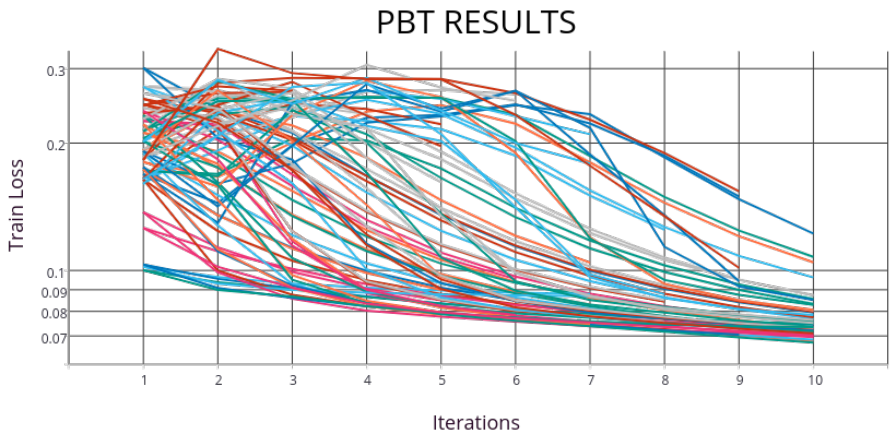


Figure 22: Visualization of hyper-parameter PBT results per experiment. 100 consecutive experiments with randomized hyper-parameter combinations were performed and the training loss is plotted against the iteration number. It can be observed that certain hyper-parameter combinations performed better than others, thus the purpose of this sweep was to find the optimal combination.

4.2 Training Results and Analysis

The model was edited to portray the optimal hyper-parameters discussed in Section 4.1 and trained for 500 epochs. The model quickly converges to the minimum training loss of -1.67673111 after only 312 epochs (as observed in Figure 23), and the model weights are saved to memory. The result of this training is a collection of weights, that when initialized in the implemented ANN architecture, result in a model that provides the lowest loss on the training dataset. Quantile-Quantile (Q-Q) plots detailing the models collective performance on the training dataset can be observed in Figures 24 to 27. Table 7 shows the Root Mean Square Error (RMSE) of the model response broken out by jammer scenario, as well as collectively.

Table 7: Model response RMSE of the collective training samples, as well as by scenario. The RMSE for the one jammer scenario is much lower than the two and three jammer cases, indicating potential over-fitting to the one jammer case.

Scenario	**Samples**	**RMSE (dBW)**
One Jammer	43,199	0.148
Two Jammer	21,296	6.522
Three Jammer	15,000	5.636
Combined	79,495	0.133

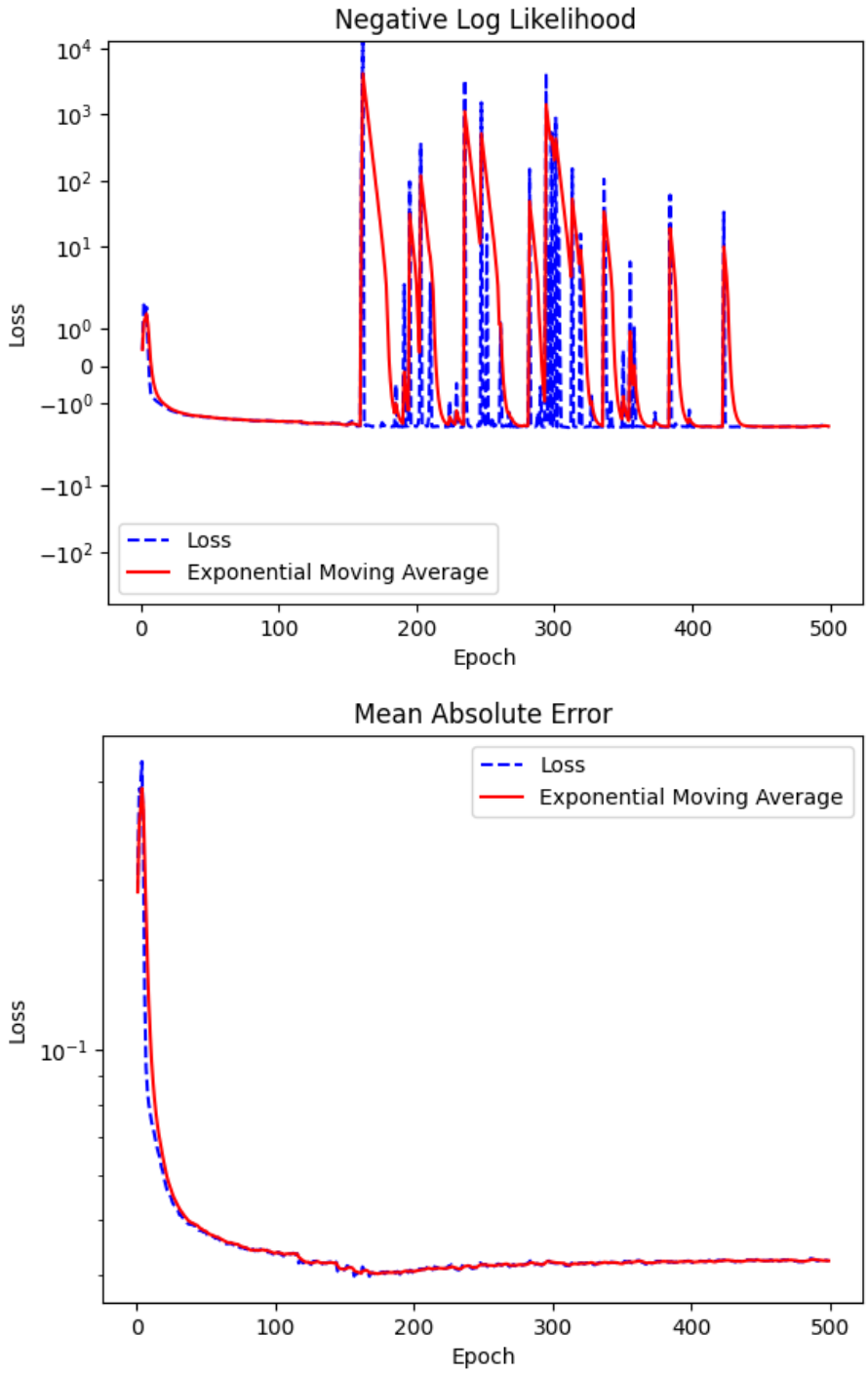


Figure 23: Negative Log Likelihood of the model response mean along with the Mean Absolute Error of the standard deviation model response. The Negative Log Likelihood plot omits the first training epoch as to avoid scaling issues. Both plots show both the loss per epoch as well as the 5 epoch moving average.

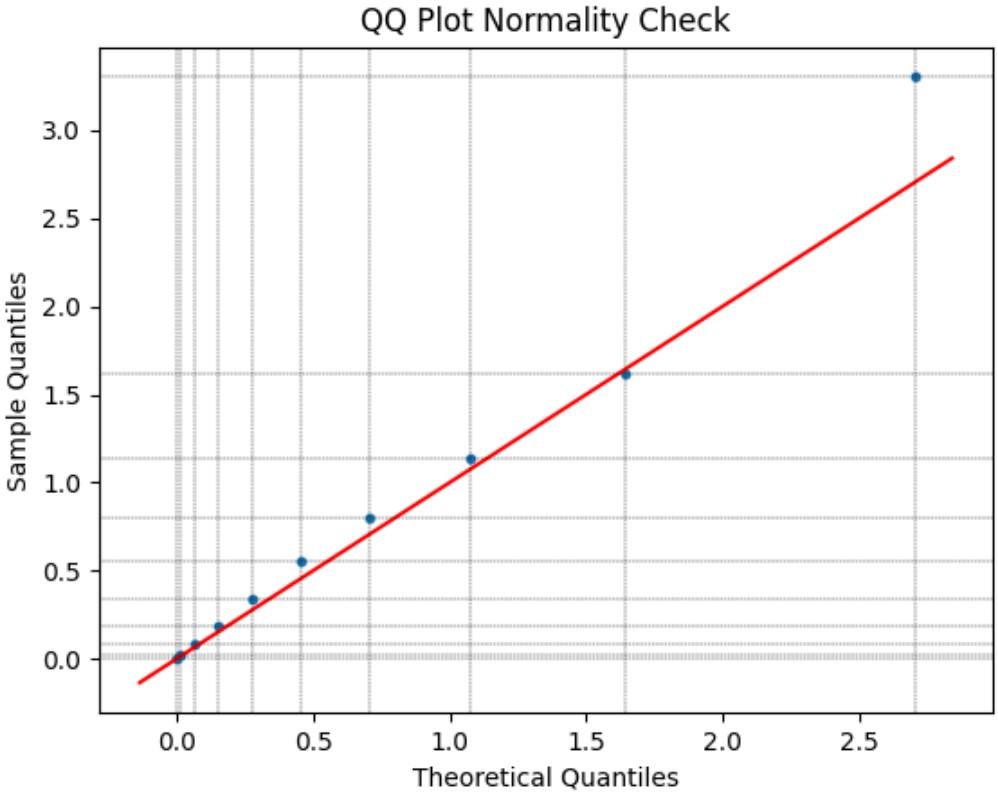


Figure 24: Q-Q plot displaying model response across all training samples. It can be observed that over the entire training set the model was conservative in its predictions until an error of about 1.6, where it became conservative. The model then became aggressive again when the error reached 2.7.

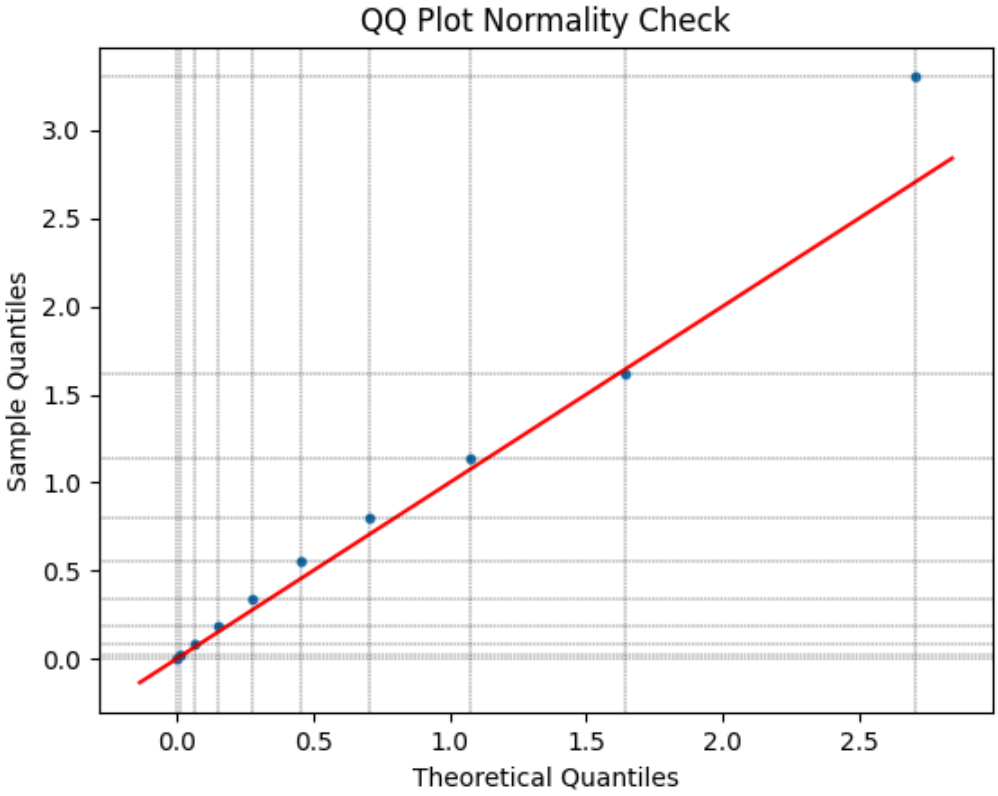


Figure 25: Q-Q plot displaying model response across all one jammer training samples. It should be noted that the model response is near identical to Figure 24, which may indicate over-fitting to the one jammer case.

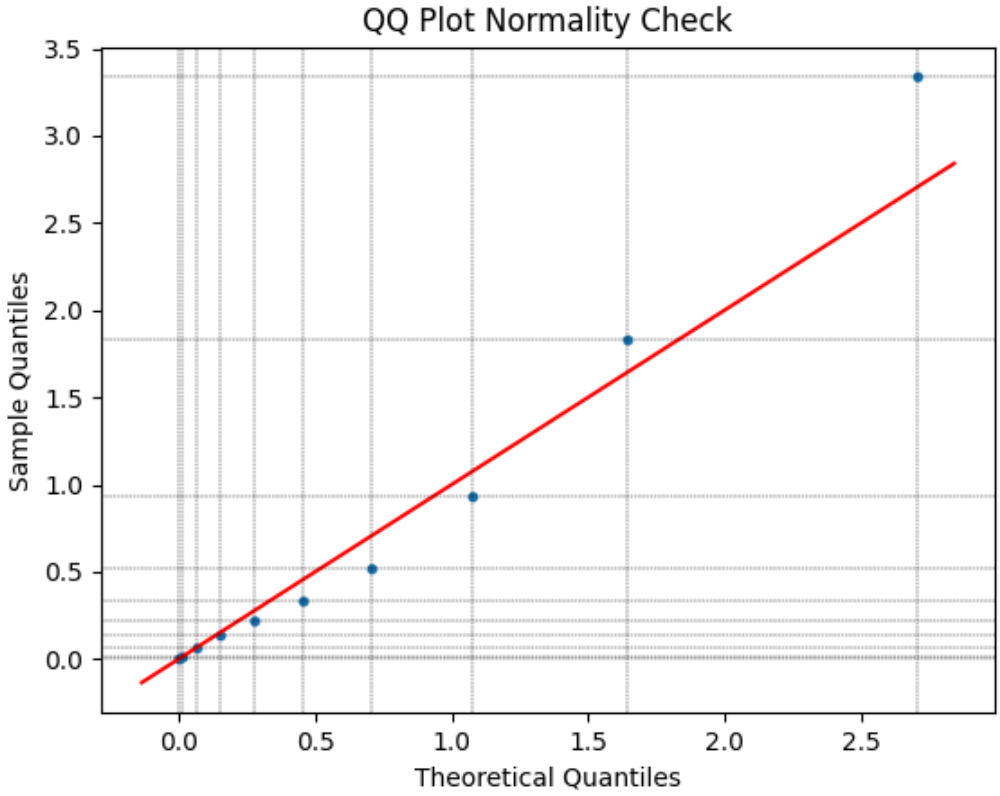


Figure 26: Q-Q plot displaying model response across all two jammer training samples. It can be observed that over the entire training set the model was aggressive in its predictions until an error of about 1.6, at which point it became conservative.

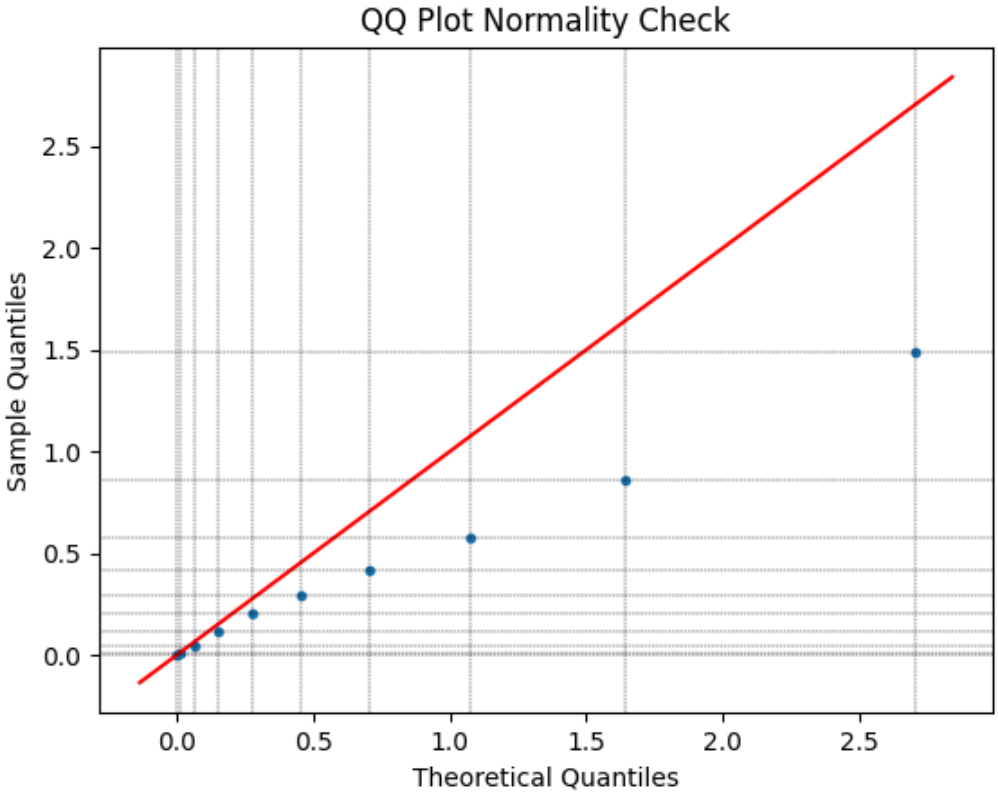


Figure 27: Q-Q plot displaying model response across all three jammer training samples. It can be observed that over the entire training set the model was aggressive in its predictions.

4.3 Test Description

Once training has been completed and the model results appeared to have generalized to the training set satisfactorily, five test data sets were generated for each of the jammer scenarios (single, double, and triple jammers). By not sequestering a test data set prior to testing, and generating a test data set post-training, we guarantee that the model has never seen the test data before. The three simulated data sets consist of one, two, and three jammers respectively, which are randomly oriented, as in a real-world scenario there is no way to know the jammer lay-down in a contested environment. (Note, the jammer lay-down, modulation scheme, and antenna parameters were generalized as to not elevate the classification of this research.) A helicopter will fly into the mission environment and traverse a fixed route, rather than a randomized one, before returning from whence it came. The test dataset was designed to mimic a helicopter flying into a contested Electromagnetic Interference (EMI) environment, losing Global Positioning System (GPS) lock, performing a data collection maneuver, and heading back to where it was deployed from. Figures 28, 34, and 40 provide a visualization of the jammer lay-downs of the test data set.

4.4 Test Results and Analysis

In order to gauge the performance of the model, there are several observables to consider. These are the true EMI environment, predicted EMI (mean of the Gaussian), and the covariance. Using these observables, the model prediction error and Mahalanobis Distance (MD) can be calculated, and a Q-Q plot can be generated. While the predicted mean and covariance will show the model's prediction of the EMI environment and its prediction confidence, the error will show how wrong the model was at each target location. The MD will show the performance adjusted for the confidence, and the Q-Q plot will aggregate many samples and show if the

model is actually parameterizing a Gaussian or not. This is turn will show whether the model is conservative or aggressive as well as any apparent bias. The following subsections will include figures for each of the described observables, but it should be noted that these are representative samples taken at random and may not represent the entire population.

4.4.1 Single Jammer Case

In this subsection we show results from the one jammer case. Figure 28 provides visualization of the true EMI environment, while Figure 29 shows a contour plot of the ANN's prediction as to what the EMI environment is for each target point. The dotted black line represents the trajectory that the helicopter followed throughout the duration of the scenario. Along this trajectory a spectrum analyzer, position, and orientation reading are sampled at 0.1 Hz and fed into the ANN described in Chapter III.

Figure 30 is a contour plot of the ANN's response variance and provides a visualization of the ANN's confidence in its response. If we plot the difference between the true EMI environment from Figure 28 and the predicted EMI environment from Figure 29 we get the ANN prediction error, as seen in Figure 31.

Once the MD is calculated for each target point the Squared MD, D_M^2, can then be calculated. After extracting the Quantiles from the D_M^2, and plotting against the Quantiles from the Chi-Squared distribution, χ^2, we obtain the Q-Q plot shown in 33.

By calculating MD such as described by Equation (13), Figure 32 can be generated, which will show the performance adjusted for confidence of the ANN.

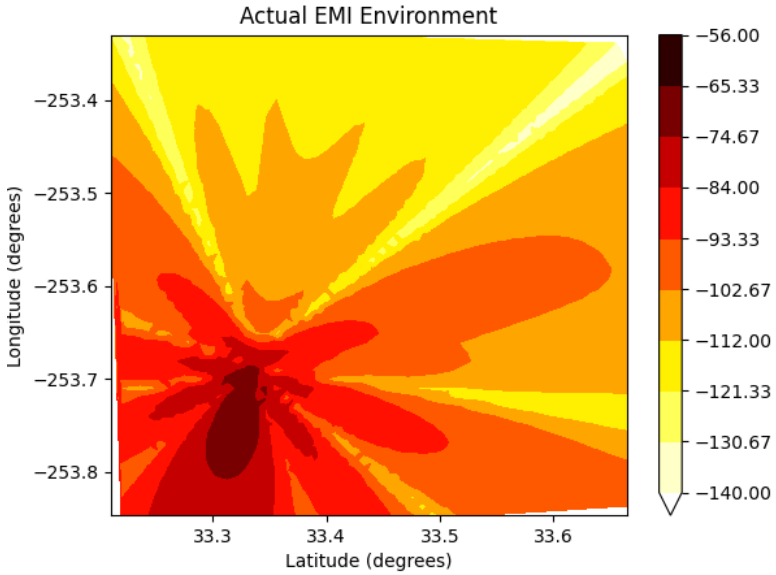


Figure 28: Single Jammer Truth EMI.

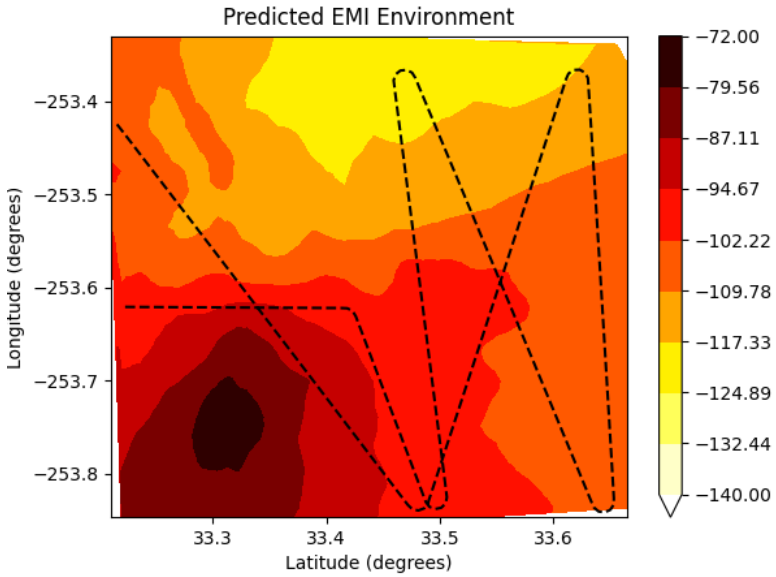


Figure 29: Tri-Contour plot providing visualization of ANNs prediction of EMI Environment in the single jammer scenario. The dotted black line is the trajectory of the helicopter throughout the scenario.

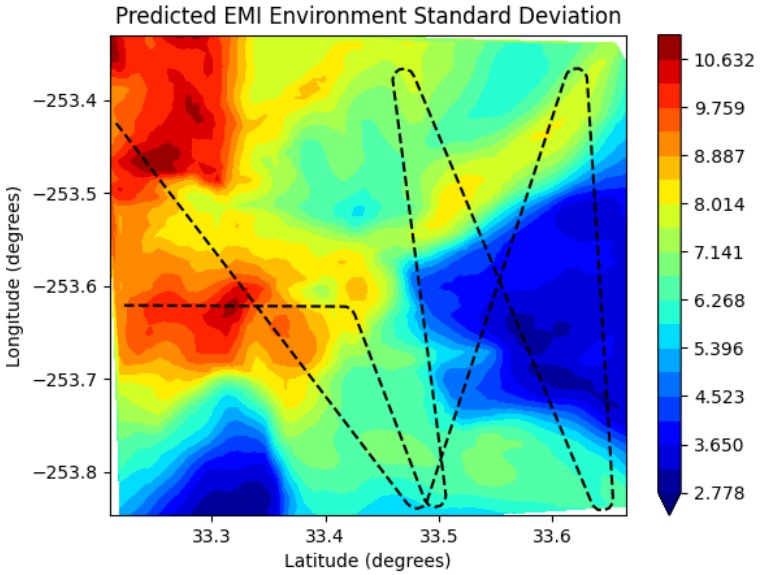


Figure 30: Tri-Contour plot providing visualization of the ANNs confidence in its response.

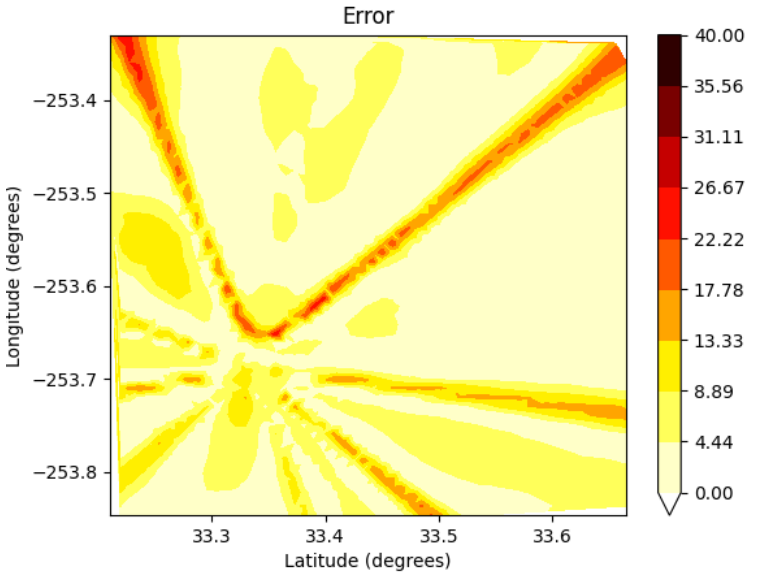


Figure 31: Single jammer ANN EMI environment prediction error magnitude.

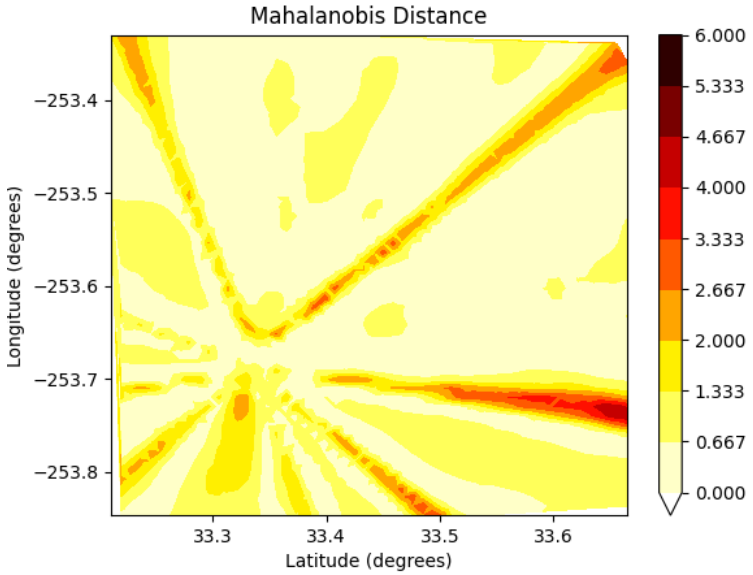


Figure 32: Single jammer ANN Mahalanobis Distance from truth.

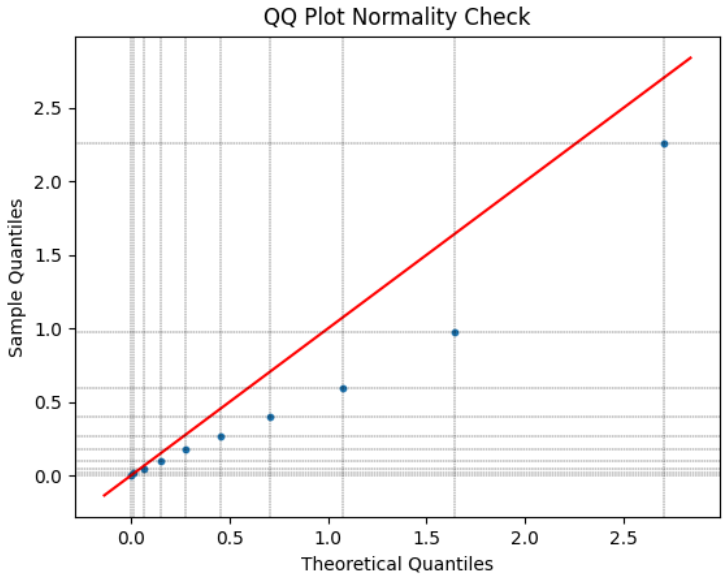


Figure 33: Single jammer Q-Q plot showing that for this single jammer sample, the model is aggressive.

By observing the true EMI environment (Figure 28) and the predicted EMI environment (Figure 29) we see that the model was able to accurately predict the EMI environment. Figure 30 displays the models variance and portrays the the model was more confident in it's prediction along the route that the helicopter traveled. The error of the models response (Figure 31) shows that the model had a difficult time trying to approximate the complex functions required to predict the jammer sidelobes. The Q-Q plot (Figure 33) allows us to ascertain that, for this single jammer scenario, the model is aggressive. This is the result of a model that was regularly over-confident in its predictions. However, this was not the case for all of the one jammer scenarios. Figure 47 shows that, over the collective one jammer training samples, the model response is fairly neutral.

4.4.2 Double Jammer Case

In this subsection we show results from the double jammer case.

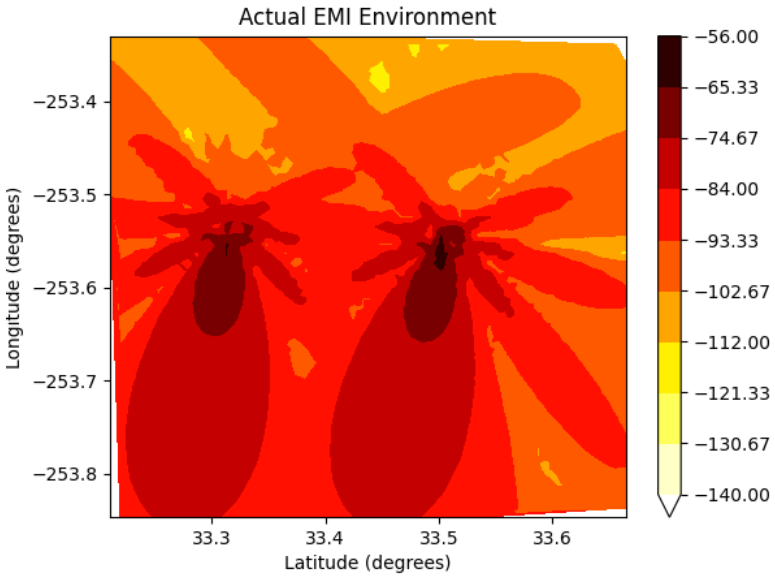


Figure 34: Double Jammer Truth EMI.

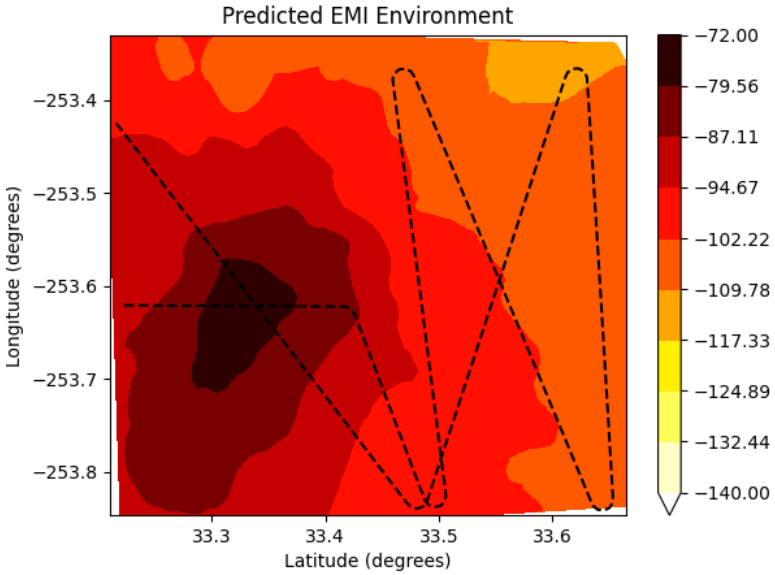


Figure 35: Tri-Contour plot providing visualization of ANNs prediction of EMI Environment for this double jammer sample. The dotted black line is the trajectory of the helicopter throughout the scenario.

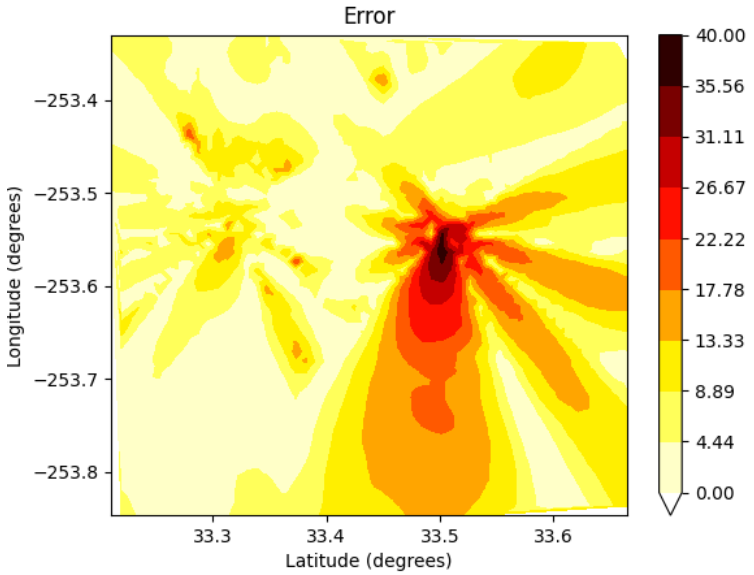


Figure 36: Double jammer ANN EMI environment prediction error.

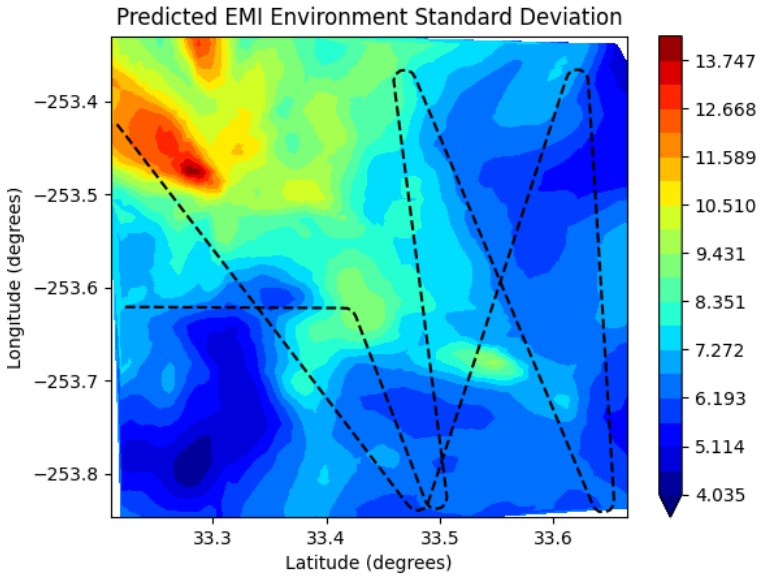


Figure 37: Tri-Contour plot providing visualization of the ANNs confidence in its response.

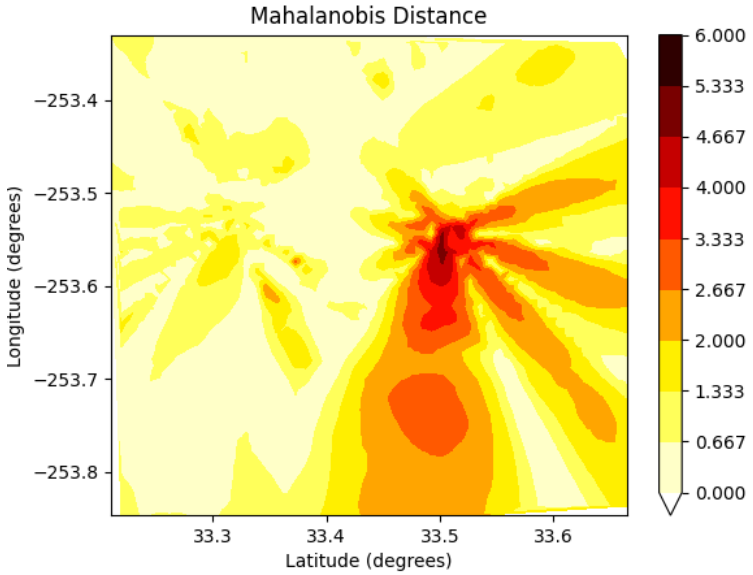


Figure 38: Double jammer ANN Mahalanobis Distance from truth.

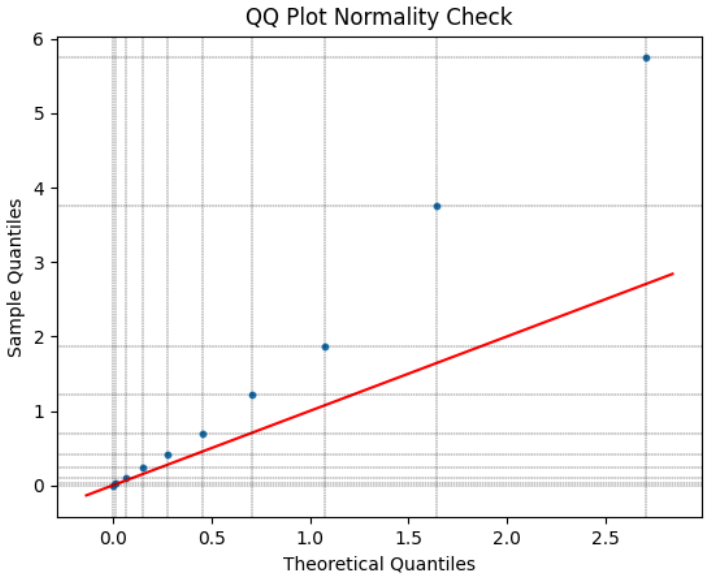


Figure 39: Double jammer Q-Q plot showing that for this double jammer sample, the model is conservative.

By observing the true EMI environment (Figure 34) and the predicted EMI environment (Figure 35) we see that the model was able to accurately predict the resultant EMI environment of only one of the two jammers. In fact, the model predicted EMI emitted by only one of the jammers, causing reason to believe that this model is actually over-fit to the single jammer dataset. Figure 37 displays the models variance and portrays the the model was more confident in it's prediction along the route that it traveled. The error of the models response (Figure 36) shows that the model had a difficult time trying to approximate the complex functions required to predict two jammers at once. The Q-Q plot (Figure 39) allows us to ascertain that, for the double jammer test data set, the model is conservative. This is the result of a model that was regularly under-confident in its predictions.

4.4.3 Triple Jammer Case

In this subsection we show results from the triple jammer case.

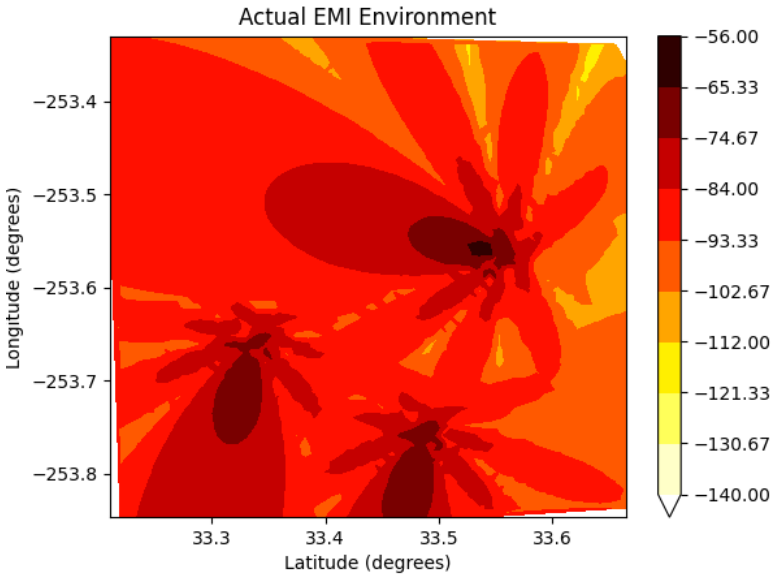


Figure 40: Triple Jammer Truth EMI.

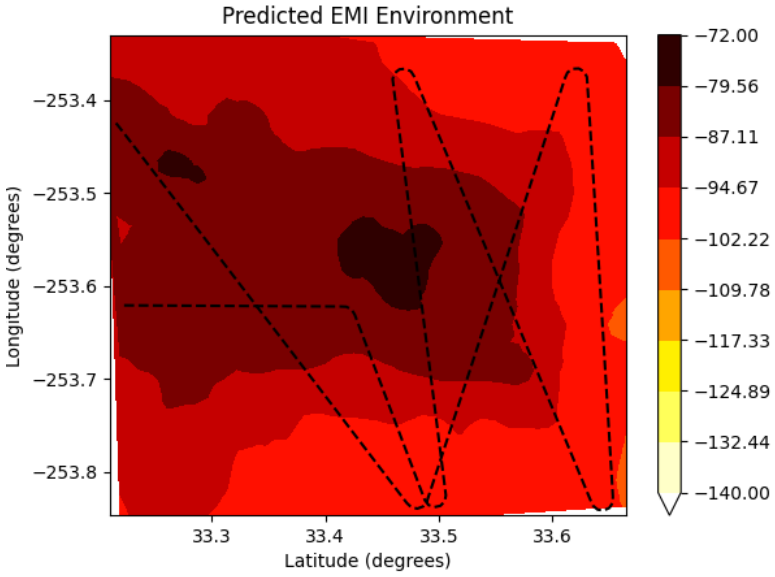


Figure 41: Tri-Contour plot providing visualization of ANNs prediction of EMI Environment for this triple jammer scenario. The dotted black line is the trajectory of the helicopter throughout the scenario.

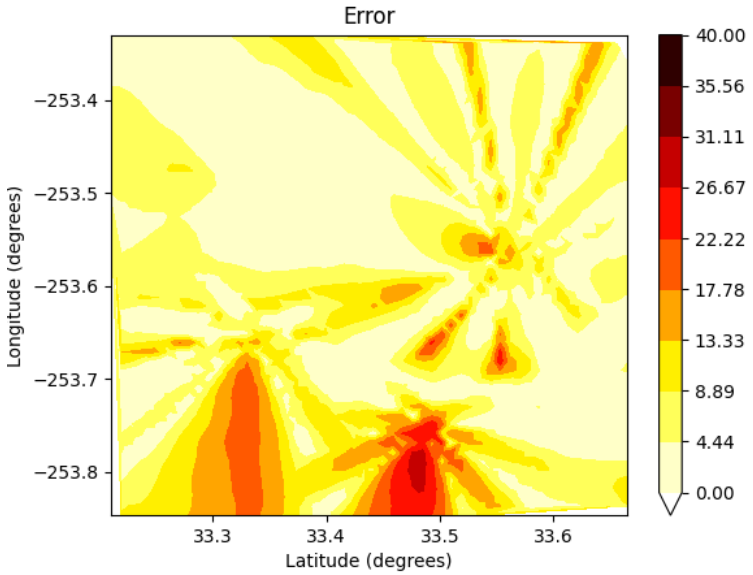


Figure 42: Triple jammer ANN EMI environment prediction error.

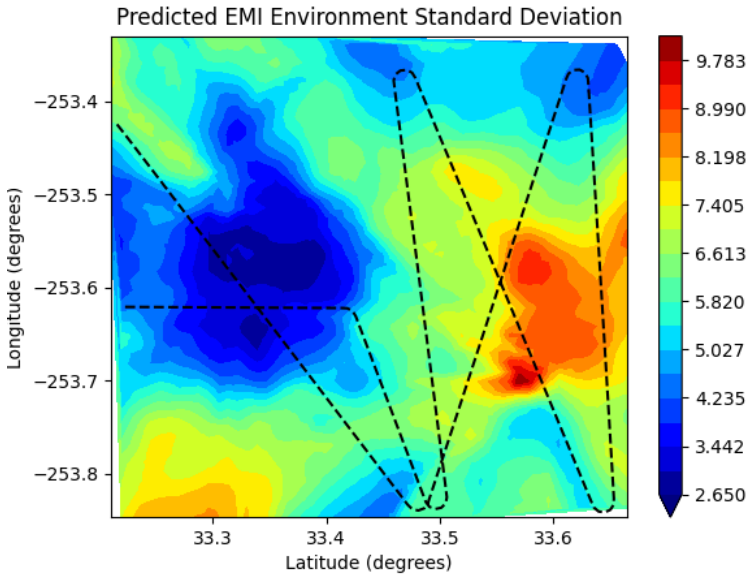


Figure 43: Tri-Contour plot providing visualization of the ANNs confidence in its response.

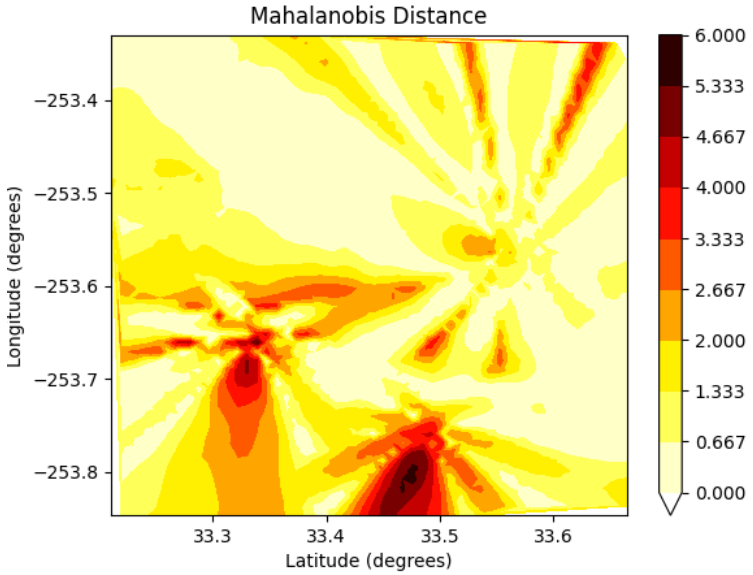


Figure 44: Triple jammer ANN Mahalanobis Distance from truth.

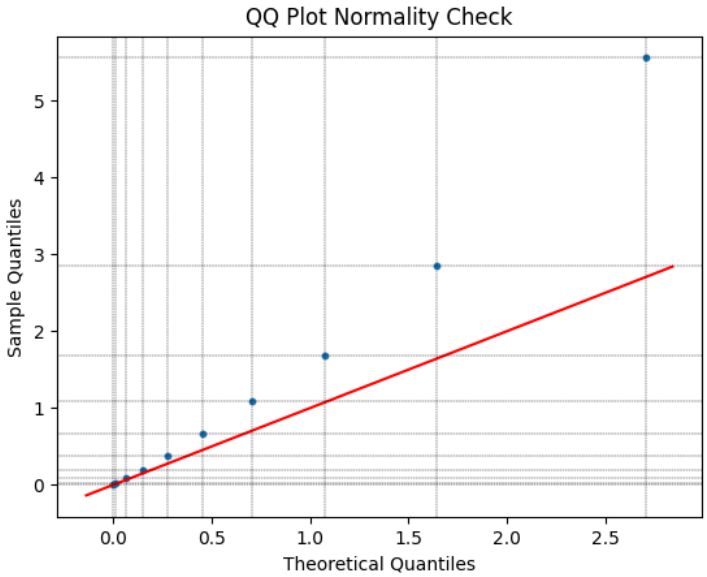


Figure 45: Triple jammer Q-Q plot showing that for this triple jammer sample, the model is conservative.

By observing the true EMI environment (Figure 40) and the predicted EMI environment (Figure 41) we see that the model was able to accurately predict the resultant EMI environment of only one of the three jammers. In fact, the model predicted EMI emitted by one jammer. However, in one of the scenarios the model actually predicted EMI emitted by two of the three jammers, providing reason to possibly dismiss the previous belief that the model is over-fit to the single jammer dataset. This likely indicates that the model requires exposure to more training data. Figure 43 displays the model's variance and portrays the the model was more confident in its prediction where it believed that the jammer was placed, rather than along the observed trajectory of the helicopter. Once again, the error of the model's response (Figure 42) shows that the model had a difficult time trying to approximate the complex functions required to predict multiple jammers simultaneously. The Q-Q plot (Figure 45) allows us to ascertain that, for the triple jammer test data set, the model is conservative. This is the result of a model that was regularly under-confident in its predictions.

4.5 Results Discussion

In the one jammer case the representative sample showed that the model response was always aggressive, while the collective samples show that the model is actually fairly neutral (as seen in Figure 47). We hypothesized that this depends on which parts of the jammer contour (main lobe, side lobes, or back lobes) were sampled for ingestion by the model. This is likely due to the model's struggle with approximating the complex functions required to predict side and back lobes. That is, the more side and/or back lobes that the model ingests the more conservative its response. To assist in approximation of these complex functions, the model could potentially benefit from increased hidden layer depth.

In the two and three jammer case the representative sample showed that the model response was always conservative. However, the collective samples from the two jammer case showed the response was mildly aggressive before turning conservative at an error of about 2.7. The collective samples from the three jammer case showed the response was consistently aggressive.

To have the two and three jammer case match the performance of the one jammer case, the model likely needs to be exposed to more training data from the two and three jammer scenarios. An interesting observation about the three jammer collective samples is that the model response is always aggressive. This could be due to the fact that the model is over-fit to the one jammer case and with three jammers oriented in such a small Area of Interest (AOI), the jamming contours closely resemble a one jammer lay-down. Table 8 shows the RMSE of the model response broken out by jammer scenario, as well as collectively. It can be observed that the combined RMSE is an order of magnitude than that of the training dataset displayed in Table 7. This indicates that the model may benefit from being exposed to additional training data to assist with generalization to the training data set.

Table 8: Model response RMSE of the collective test samples, as well as by scenario. The combined RMSE of the test data set was an order of magnitude higher than that of the training data set, indicating that additional steps must be taken to assist model generalization to the training dataset.

Scenario	**Samples**	**RMSE (dBW)**
One Jammer	5	2.987
Two Jammer	5	3.137
Three Jammer	5	2.894
Combined	15	1.676

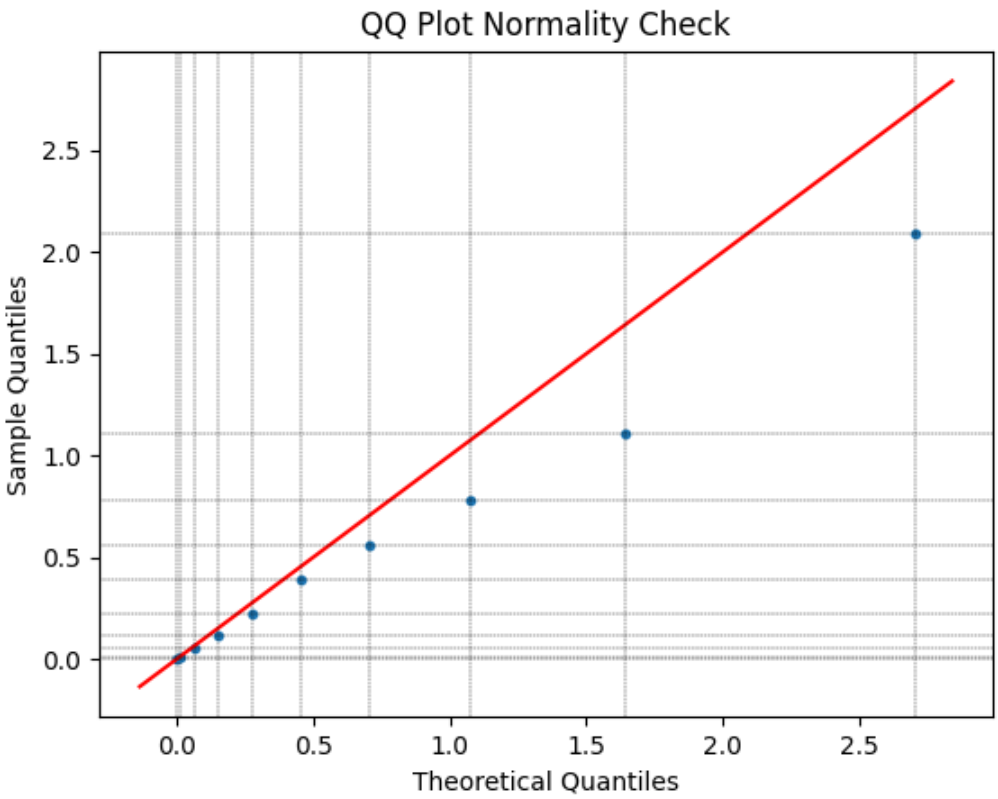


Figure 46: Q-Q plot displaying model response across all test samples. It can be observed that over the entire training set the model was aggressive in its predictions.

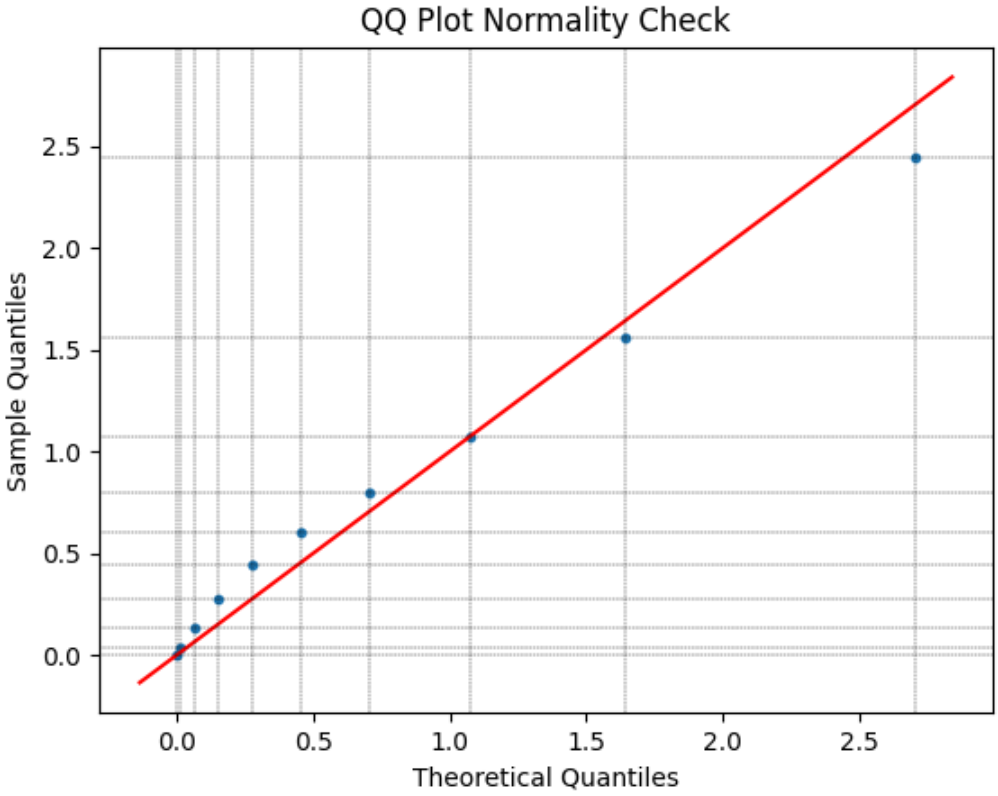


Figure 47: It can be observed that over the entire one jammer test set the model response was fairly neutral. Predictions were mildly conservative until an error of about 1.0, at which point it became mildly aggressive.

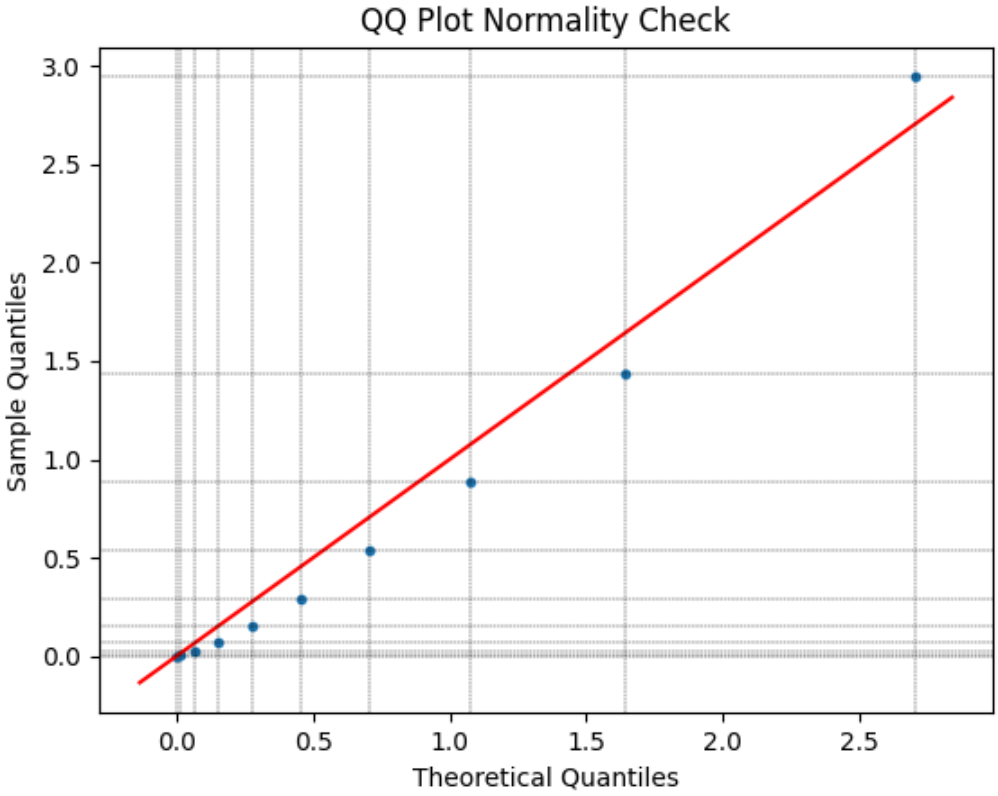


Figure 48: It can be observed that over the entire two jammer test set the model response was mildly aggressive until an error of about 2.7, at which point it became mildly conservative.

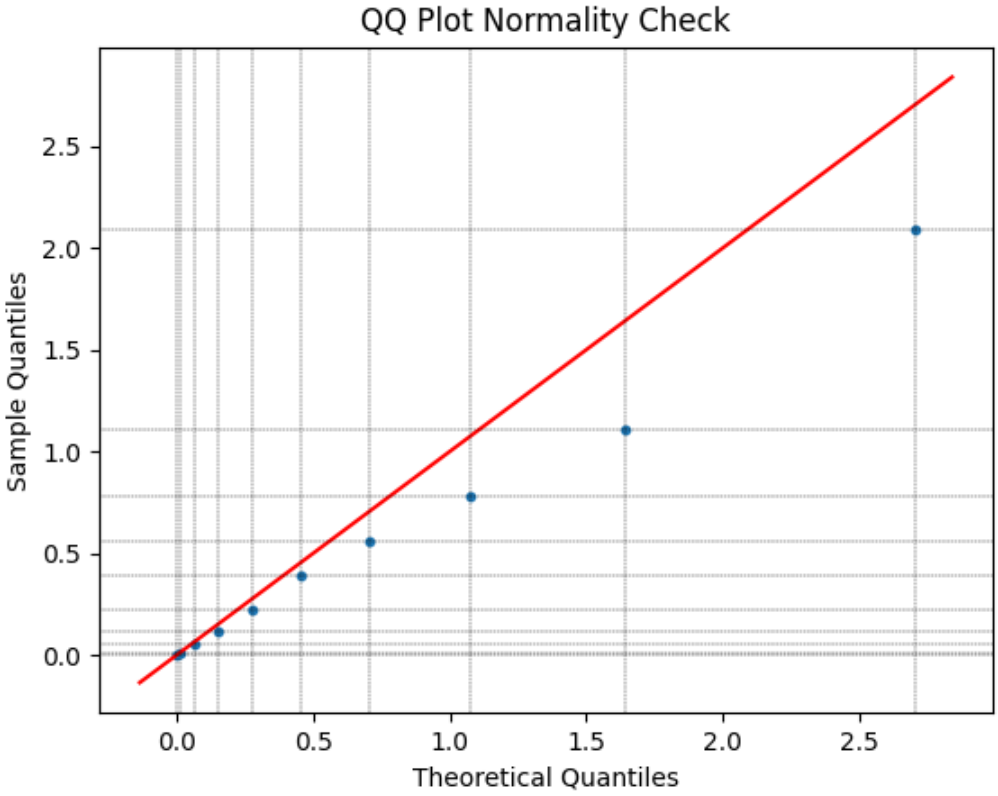


Figure 49: It can be observed that over the entire three jammer test set the model response was continuously aggressive. This could be due to the model over-fitting to the one jammer case.

V. Conclusions

In conclusion, we have presented a problem in which a helicopter is traversing a predefined route before finding itself amid a contested Electromagnetic Interference (EMI) environment. An Artificial Neural Network (ANN) was developed and trained which used vehicle dynamics and spectrum analyzer features in order to predict the EMI environment given only a small subset of points in the mission environment. By using the flexibility of Conditional Neural Processes (CNPs), and the ability to extract prior knowledge from training data, we have demonstrated the ability to perform prediction of the EMI environment without the need of a distributed network of monitoring nodes.

It has been shown that the model developed in this thesis, while it could use improvements, has the capability to predict the EMI environment as well as providing a measure of confidence in the prediction. For the one jammer case the model was observed to behave fairly neutrally. The model response for the two jammer case was aggressive before the error reached 2.7, then it became conservative. The three jammer case model response was consistently aggressive. The single, double, and triple jammer test set had a Root Mean Square Error (RMSE) of 2.988 dBW, 3.137 dBW, and 2.894 dBW respectively and a RMSE of 1.676 dBW collectively.

In the following section we propose multiple ideas that could aid in the improvement of the model.

5.1 Future Work

For future work the first proposal that would benefit this research would be elevating the security classification level of this research and use any and all jammer specifications and lay-downs that our intelligence community may have gathered in

the past. Second, adding extra input features such as: additional receive antennas, multi-frequency spectrum analyzer readings, and the conversion of helicopter's position from LLA to Local-Level coordinates may benefit the ANN. Third, this research would likely benefit from a much larger dataset, increased ANN depth, and more extensive hyper-parameter sweeps, ideally massaging the ANN to the point where it is able to learn more complex approximation functions to better model the side lobes of the interference sources. Fourth, fewer assumptions when simulating the data would provide more realistic training features. That is, it should not be assumed there is no antenna or aircraft body shading, or that there are no terrain effects. Fifth, the test set would ideally use non-simulated features from an actual Joint Navigation Warfare Center (JNWC) Joint Urgent Operational Need (JUON) test event to test the model. With these future work propositions implemented, this thesis work would move closer to the point where a machine learning solution could be implemented to solve the JUON CC-0575.

Bibliography

1. Defense Science Board. Fulfillment of Urgent Operational Needs. Technical report, Defense Science Board, Washington, D.C., 2009.

2. Per Eng and Pratap Misra. *Global Positioning System Signals, Measurements, Performance*. Jamuna Press, Lincoln, revised se edition, 2020.

3. National Coordination Office for Space-Based Positioning, Navigation, and Timing. Space Segment, 2020. `https://www.gps.gov/systems/gps/space/`.

4. National Coordination Office for Space-Based Positioning, Navigation, and Timing. Control Segment, 2018. `https://www.gps.gov/systems/gps/control/`.

5. John A. Volpe National Transportation Systems Center. Vulnerability assessment of the transportation infrastructure relying on the global positioning system. Technical report, U.S. Department of Transportation, Washington, D.C., 08 2001.

6. Philip Kwan. NAVSTAR GPS Space Segment/Navigation User Segment Interfaces. *IS-GPS-200K*, 2019. `https://www.gps.gov/technical/icwg/IS-GPS-200K.pdf`.

7. R Kalafus. Interference to GPS Receivers from Mobile Satellite Emissions. *Proceedings of the 11th International Technical Meeting of the Satellite Division of the Institute of Navigation GPS-98*, pages 15–18, 1998.

8. P Ward. GPS Receiver RF Interference Monitoring, Mitigation, and Analysis Techniques. *Journal of the Institute of Navigation*, 41(4), 1994.

9. Tim Erbes. Protecting Downstream GPS Systems from Jamming and Spoofing. In *Proceedings of the ION 2018 Joint Navigation Conference*, 2018.

10. Navisp. GIDAS (GNSS INTERFERENCE DETECTION & ANALYSIS SYSTEM), 2020.

11. Chollet Francois. *Deep Learning with Python*. Manning, Shelter Island, 2018.

12. Daniela Witte, Trevor Hastie, Gareth James, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, New York, 7 edition, 2017.

13. Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

14. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

15. Marta Garnelo, Dan Rosenbaum, Chris J. Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. Rezende, and S. M. Ali Eslami. Conditional neural processes, 2018.

16. Wikipedia: Systems Tool Kit, 2020. `https://en.wikipedia.org/wiki/Systems_Tool_Kit`.

17. AGI. STK Programming Help, 2020. `https://help.agi.com/stkdevkit/index.htm#DevelopmentEnvironments.htm%3FTocPath%3DDevelopment%2520Environments%7C_____0`.

18. Saurabh Singh. Elu as an activation function in neural networks, June 2020.

19. Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018. `https://arxiv.org/abs/1807.05118`.

20. P.C Mahalanobis. On the Generalised Distance in Statistics. *Proceedings of the National Institute of Sciences of India*, pages 49–55, 1936.

21. R. De Maesschalck, D. Jouan-Rimbaud, and D.L Massart. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18, 2000.

22. MATLAB. *Statistics and Machine Learning Toolbox Version 11.7 (R2020a)*. The MathWorks Inc., Natick, Massachusetts, 2020.

23. M.B. Wilk and R. Gnanadesikan. Probability Plotting Methods for the Analysis of Data. *Biometrika*, 55(1):1–17, 1968.

Acronyms

AFIT Air Force Institute of Technology. 32

AGI Analytical Graphics, Inc. 18

AI Artificial Intelligence. 11

ANN Artificial Neural Network. iv, 1, 2, 3, 13, 14, 15, 16, 17, 18, 21, 29, 39, 41, 42, 50, 69, 70, 1

AOI Area of Interest. 22, 28, 63

CDN Cyber Development Network. 32

CNP Conditional Neural Process. iv, 2, 18, 21, 29, 69, 1

COP Common Operating Picture. iv, 1, 2, 1

CW Continuous Wave. 8

dBi Decibel Isotropic. 22

dBW Decibel Watts. 21, 22

DoD Department of Defense. 3

EA Electronic Attack. 10

EGI Embedded Global Positioning System (GPS) and Inertial Navigation System (INS). 22

ELU Exponential Linear Unit. 32

EMI Electromagnetic Interference. iv, vii, 1, 2, 24, 25, 26, 27, 32, 34, 36, 49, 50, 54, 58, 62, 69, 1

EP Electronic Protection. 10

ES Electronic Support. 10

FOM Figure of Merit. 28

GIDAS GNSS Interference Detection and Analysis System. 11

GNSS Global Navigation Satellite System. 10

GP Gaussian Process. 18

GPS Global Positioning System. iv, vii, xi, 1, 3, 4, 5, 6, 7, 8, 9, 10, 22, 49, 74, 1

IF Intermediate Frequency. 8

INS Inertial Navigation System. 22, 74

JNWC Joint Navigation Warfare Center. 1, 70

JUON Joint Urgent Operational Need. iv, 1, 70, 1

LHCP Left-Hand Circularly Polarized. 7

MCS Master Control Station. 5

MD Mahalanobis Distance. viii, 21, 34, 35, 36, 39, 49, 50

MEO Medium Earth Orbit. 6

ML Machine Learning. iv, 1, 3, 11, 12, 13, 14, 1

MSE Mean Squared Error. 17

NAVWAR Navigation Warfare. 10

NLL Negative Log Likelihood. 32, 34

PBT Population Based Training. viii, 33, 41, 42

PNT Position Navigation and Timing. 6, 10

Q-Q Quantile-Quantile. viii, ix, x, 39, 40, 42, 45, 46, 47, 48, 49, 50, 53, 54, 57, 58, 61, 62, 65

RF Radio Frequency. xi, 4, 8, 9, 19, 21

RHCP Right-Hand Circularly Polarized. 7

RMSE Root Mean Square Error. xi, 42, 43, 63, 64, 69

SPS Standard Positioning Service. 10

STK Systems Tool Kit. xi, 18, 19, 20, 21, 22, 28, 29

SV Space Vehicle. 5

USAF United States Air Force. 9

WSMR White Sands Missile Range. 27

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
27–11–2020	Master's Thesis	Jan 2019 — Oct 2020

4. TITLE AND SUBTITLE

Electromagnetic Interference Estimation via Conditional Neural Processing

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Edgar E. Gomez

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-20-D-006

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

XXXX/XXXX

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of this thesis is to determine the efficacy of employing ML to solve JUON CC-0575, which aims to develop a COP of the GPS EMI environment. With the growing popularity of ANNs, ML solutions are quickly gaining traction in businesses, academia and government. This in turn allows for problem solutions that were previously inconceivable using the classical programming paradigm. This thesis proposes a method to develop a COP of the battlefield via ANN ingestion of multiple-source signals and sensors.
We conduct three separate experiments with varying amounts of EMI interference sources. The type of ANN developed to address this problem is a CNP with residual connections. The model is developed to provide the estimated EMI environment as well as a measure of confidence in its estimates, as the specific application of this model could lead to loss of life in the event the model estimates are taken as truth.

15. SUBJECT TERMS

subject terms here

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Major Joseph A. Curro, AFIT/ENG
U	U	U	UU	89	19b. TELEPHONE NUMBER *(include area code)* (937) 255-6565, ext 4620; joseph.curro@afit.edu